Tema 5. Seguridad Perimetral

Administración de Redes

Índice

- Introducción
- Tecnologías Firewall
  - Filtrado de paquetes estático
  - Filtrado dinámico de paquetes
  - Filtrado en capa de aplicación
  - NAT/NATP
- Seguridad Perimetral: Políticas

1. Filtrado estático:
Aplica reglas básicas (IP, puerto, protocolo). Es rápido pero menos seguro.

2. Filtrado dinámico:
Recuerda el estado de las conexiones (al recordar las conexiones permite responder automaticamente). Más seguro que el estático.

3. Filtrado en capa de aplicación:
Inspecciona el contenido de los paquetes (ej. HTTP, FTP). Muy seguro, pero más lento.

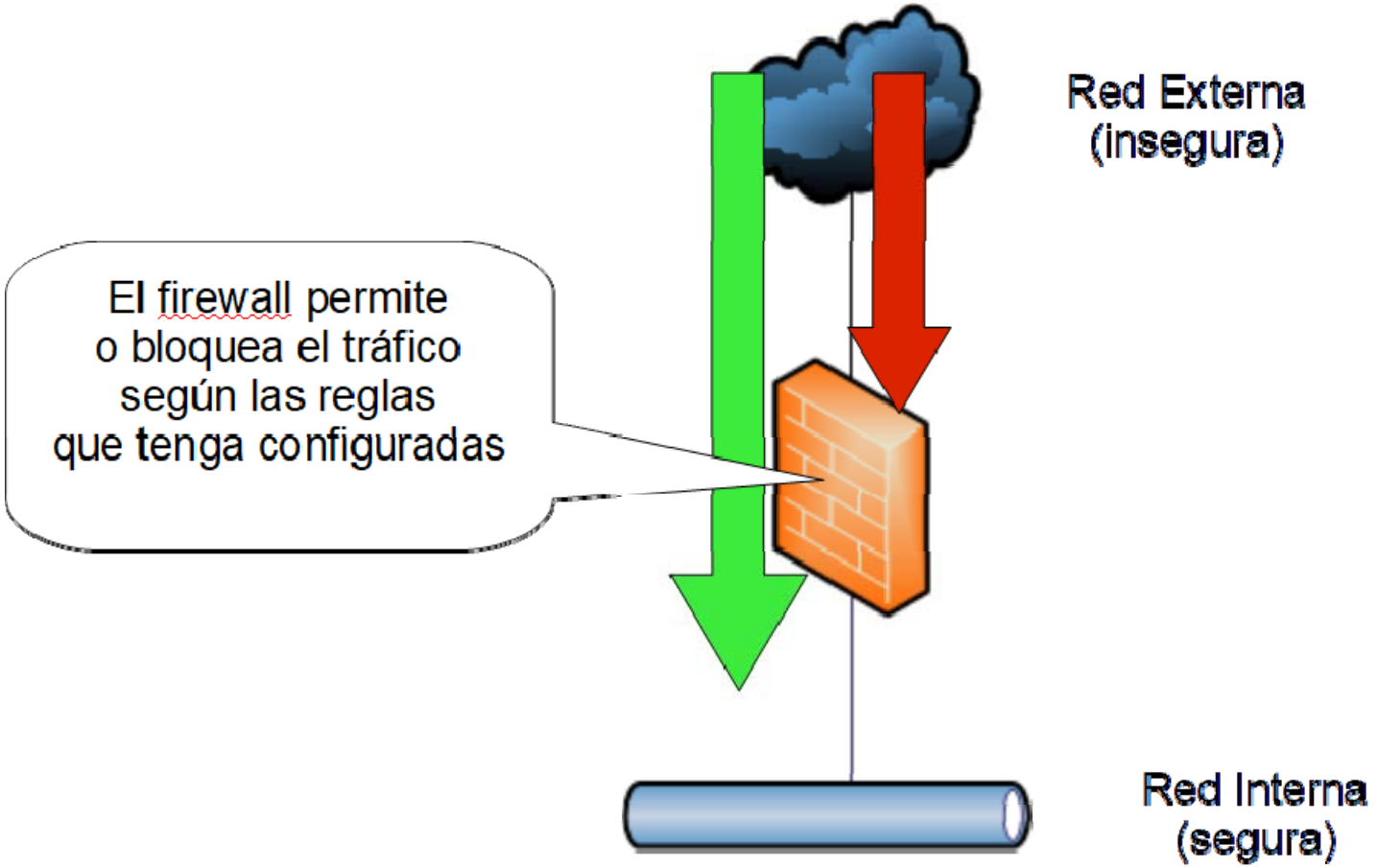
4. NAT/NATP:
Oculta las IP internas y permite compartir una IP pública. Añade seguridad.

1- Introducción

Introducción

- Firewalls: Son dispositivos que filtran el tráfico entre redes
  - Distintos requisitos de seguridad en cada red
  - Funciones: analizar, registrar y, posiblemente, bloquear en parte el tráfico de red
  - Pueden actuar en distintas capas do modelo OSI

Introducción



Introducción

- Aportaciones:
  - Ayuda a implementar políticas de control de acceso
  - Es la primera línea de defensa, que bloquea un buen número de ataques
  - Minimiza los problemas provocados por las vulnerabilidades conocidas de protocolos y servicios, al limitar la conectividad entre redes
    - Exposición de servicios internos y vulnerables como NFSv3 o SMB, fallos de implementación de la pila TCP/IP, fallos de implementación en servidores.
  - Ayuda a implementar la política de seguridad:
    - Las reglas del firewall se establecen en base a la política de seguridad de la organización, donde se establece qué tráfico permitir y cual no, entre las diferentes redes
  - Permiten guardar un registro (*log*) de tráfico de red

Introducción

- Limitaciones
  - Ineficaces ante *bugs* en aplicaciones o servicios permitidos
  - No protege contra ataques mediante conexiones autorizadas
    - Cuidado con las conexiones desde el interior
  - ¡Sólo protegen frente a conexiones que pasan por él!
    - Accesos alternativos (e.g.: redes móviles)
    - Accesos desde dentro (e.g.: Wifi mal protegida, *malware* en memorias USB, etc.)
  - Pueden llegar a ser molestos para los usuarios
  - Deben configurarse y administrarse cuidadosamente
  - Deben ser debidamente monitorizados

:no puede evitar ataques que explotan fallos (bugs)

no puede detectar ni bloquear ataques que vienen a través de conexiones que ya están permitidas y autorizadas.

Si un acceso no pasa por el firewall (como una red móvil o wifi insegura), el firewall no puede protegerte.

- ¡Son una línea más de defensa, pero no deberían ser la única!

2- Tecnologías Firewall

Introducción

- Los firewalls se suelen clasificar en base a al tecnología que utilizan para tomar las decisiones de reenvío y filtrado
- Las técnicas clásicas son:
  - Firewalls de filtrado de paquetes
    - Filtrado de paquetes estático
    - Filtrado de paquetes con estado o dinámico
  - Firewalls a nivel de aplicación

- Además es importante, a la hora de considerar el filtrado de tráfico, considerar el comportamiento del direccionamiento IPv4 con NAT, por lo que se abordará esta tecnología como parte de este capítulo

- Por último, se presentarán también los firewalls basados en zonas y los de nueva generación

Tecnologías Firewall

2.1- Firewall de Filtrado Estático de Paquetes

Filtrado Estático de Paquetes

: Es una técnica usada en firewalls para controlar qué tráfico entra o sale de una red, basándose en reglas simples.

- Operan normalmente en la capa 3 y 4 del modelo OSI Red y transporte
- Filtrado en base a características de la cabecera del paquete IP : matching (el equipo analiza las características del paquete).
  En el filtrado estático comprueba algunas de las características que aparecen ahí

  Las características son las que nosotros le especifiquemos unicamente en la regla..

  IP
  - Dirección IP de origen
  - Dirección IP de destino
  - Tipo de tráfico (TCP, UDP, ICMP)
  - Ciertas características de capa 4:

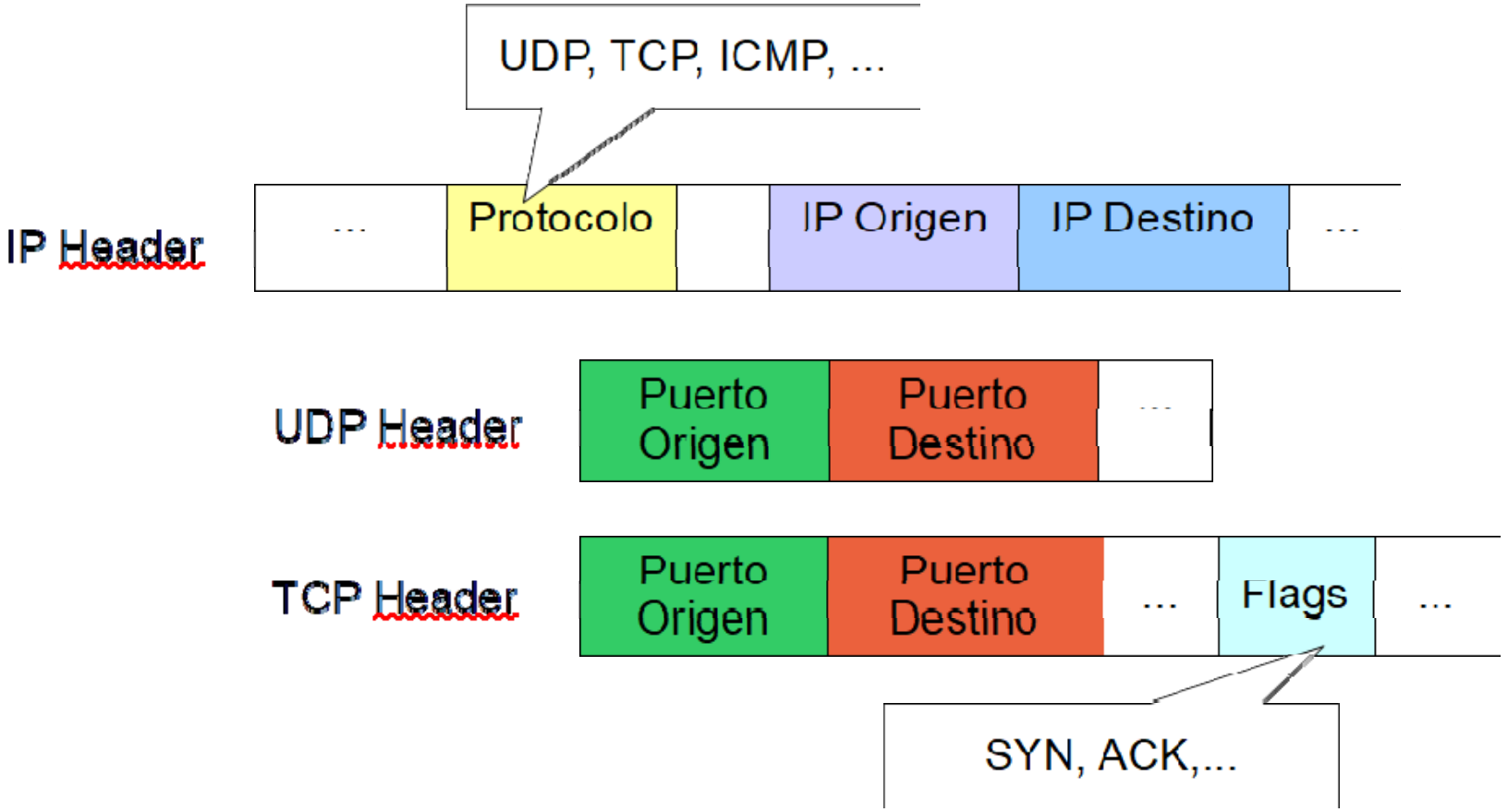
    PROTOCOLO
    - En UDP, puertos origen y destino
    - En TCP, puertos origen y destino, *flags*
    - En ICMP, comando / respuesta
  - Interfaz de red por la que llega o se envía el paquete

    PUERTO

- **No guardan información del contexto** : NO recuerda conexiones anteriores
  - Se analiza cada paquete individualmente

¿Porque es estático?
Porque las reglas no cambian automáticamente y no guarda información del estado de las conexiones. Cada paquete se analiza como si fuera único.

Filtrado Estático de Paquetes



Filtrado Estático de Paquetes

- **Dirigidos por un conjunto de reglas**

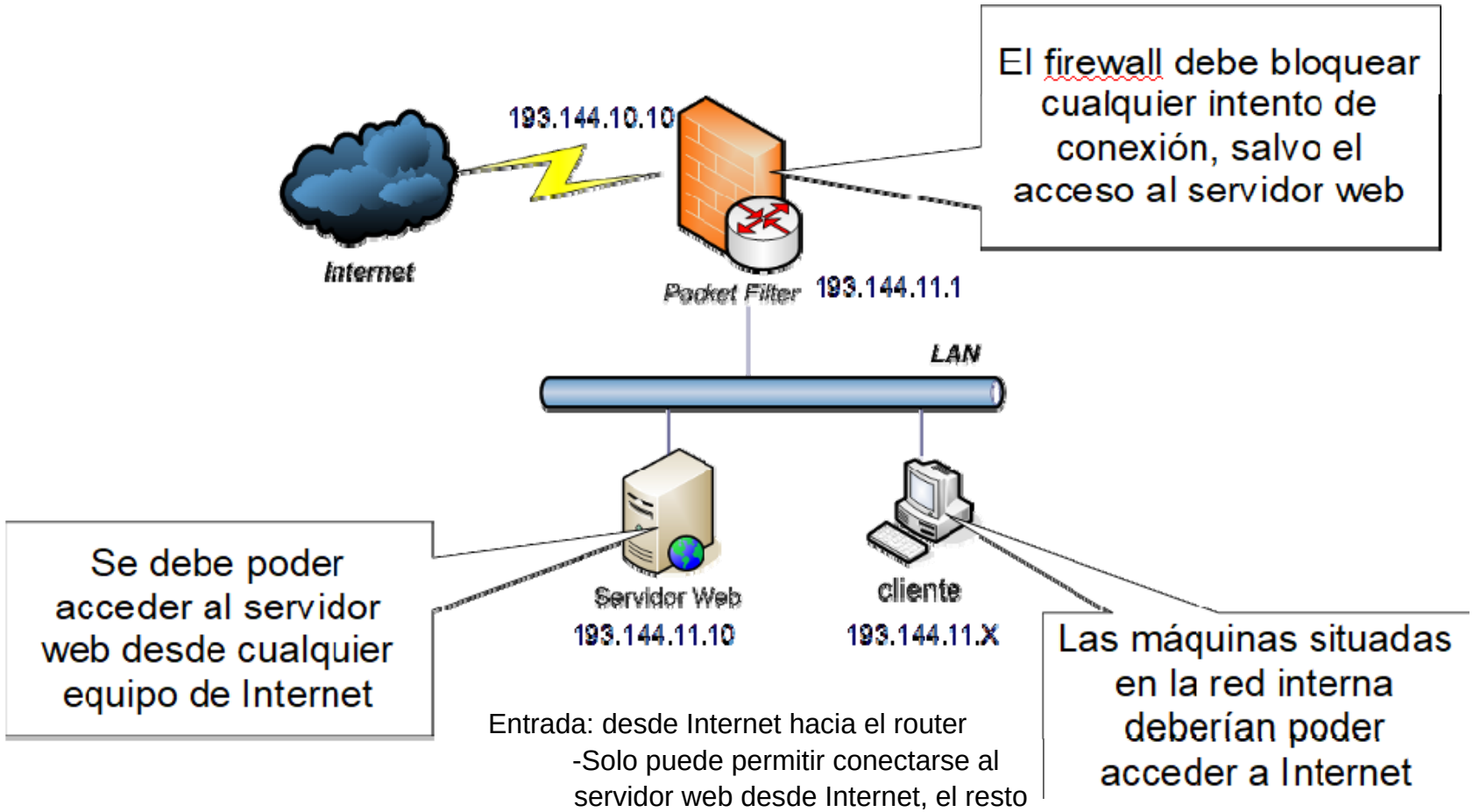
1-PARTE DE MATCHING: Una parte de *matching*, con la que deben coincidir campos de la cabecera del paquete para que se aplique la regla
  - Sentido del paquete: entrada/salida
  - Dirección IP de origen/destino, tipo de tráfico, puerto, etc...

-> Si el paquete coincide con esto, se pasa a la acción.

2-ACCIÓN A REALIZAR: Acción a realizar
  - Aceptar: El paquete pasa el firewall
  - Denegar: El paquete se descarta y se notifica al origen
  - Descartar: El paquete se descarta sin informar al origen
  - Acciones adicionales: e.g. “logging” Registrar (opcional)

- **Funcionamiento: Las reglas se procesan en orden**
  - Se examinan los campos indicados en la parte de *matching*
  - Si el paquete coincide se lleva a cabo la acción indicada
  - Si no se continúa con la siguiente regla

Filtrado Estático de Paquetes



Entrada: desde Internet hacia el router
-Solo puede permitir conectarse al servidor web desde Internet, el resto se bloquea

Salida: controla lo que sale desde la red interna hacia fuera. Permite solo salir a la ip de la red pero en el puerto 80 como destino.

Filtrado Estático de Paquetes

```
! Reglas de entrada en la interfaz externa
FW(config)# ip access-list extended entrada
! Deniega IP-spoofing
FW(config-ext-nacl)# deny ip 193.144.11.0 0.0.0.255 any
! Deniega conexiones al propio firewall
FW(config-ext-nacl)# deny ip any host 193.144.10.10
FW(config-ext-nacl)# deny ip any host 193.144.11.1
! Permite acceso al servidor web
FW(config-ext-nacl)# permit tcp any host 193.144.11.10 eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any
```

//se crea una ACL Extendida para lo que ENTRA al router: solo puede se puede conectar al server web desde Internet. El resto se bloquea.

```
! Reglas de salida en la interfaz externa
FW(config-ext-nacl)# ip access-list extended salida
! Permitir equipos de la LAN acceder a Internet
FW(config-ext-nacl)# permit tcp 193.144.11.0 0.0.0.255 any eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any
```

//se crea una ACL Extendida para lo que SALE del router: solo se puede conectar a Internet el cliente y bloquea todo lo demás.

Filtrado Estático de Paquetes

```
! Reglas de entrada en la interfaz externa
FW(config)# ip access-list extended entrada
! Deniega IP-spoofing
FW(config-ext-nacl)# deny ip 193.144.11.0 0.0.0.255 any
! Deniega conexiones al propio firewall
FW(config-ext
FW(config-ext
! Permite acc
FW(config-ext
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any

! Reglas de salida en la interfaz externa
FW(config-ext-nacl)# ip access-list extended salida
! Permitir equipos de la LAN acceder a Internet
FW(config-ext-nacl)# permit tcp 193.144.11.0 0.0.0.255 any eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any
```

(1) Denegar paquetes entrantes con IP de origen en la LAN (evita ataques IP-Spoofing)

Denegar paquetes que vienen de afuera pero que dicen tener IP de la red interna para evitar que un atacante finge ser alguien dentro de la red (eso es IP spoofing).

Filtrado Estático de Paquetes

```
! Reglas de entrada en la interfaz externa
FW(config)# ip access-list extended entrada
! Deniega IP-spoofing
FW(config-ext-nacl)# deny ip 193.144.11.0 0.0.0.255 any
! Deniega conexiones al propio firewall
FW(config-ext-nacl)# deny ip any host 193.144.10.10
FW(config-ext-nacl)# deny ip any host 193.144.11.1
! Permite acceso al servidor
FW(config-ext-nacl)# pe
! Denegar todo por defe
FW(config-ext-nacl)# de

! Reglas de salida en la interfaz externa
FW(config-ext-nacl)# ip access-list extended salida
! Permitir equipos de la LAN acceder a Internet
FW(config-ext-nacl)# permit tcp 193.144.11.0 0.0.0.255 any eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any
```

(2) y (3) Denegar paquetes entrantes con destino el propio FW

Filtrado Estático de Paquetes

```
! Reglas de entrada en la interfaz externa
FW(config)# ip access-list extended entrada
! Deniega IP-spoofing
FW(config-ext-nacl)# deny ip 193.144.11.0 0.0.0.255 any
! Deniega conexiones al propio firewall
FW(config-ext-nacl)# deny ip any host 193.144.10.10
FW(config-ext-nacl)# deny ip any host 193.144.11.1
! Permite acceso al servidor web
FW(config-ext-nacl)# permit tcp any host 193.144.11.10 eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any

! Reglas de salida en la interfa
FW(config-ext-nacl)# ip access-l
! Permitir equipos de la LAN acce
FW(config-ext-nacl)# permit tcp 193.144.11.0 0.0.0.255 any eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any
```

(4) Permitir paquetes TCP entrantes destinados al servidor web y al puerto 80

Filtrado Estático de Paquetes

```
! Reglas de entrada en la interfaz externa
FW(config)# ip access-list extended entrada
! Deniega IP-spoofing
FW(config-ext-nacl)# deny ip 193.144.11.0 0.0.0.255 any
! Deniega conexiones al propio firewall
FW(config-ext-nacl)# deny ip any host 193.144.10.10
FW(config-ext-nacl)# deny ip any host 193.144.11.1
! Permite acceso al servidor web
FW(config-ext-nacl)# permit tcp any host 193.144.11.10 eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any

! Reglas de salida en la i
FW(config-ext-nacl)# ip a
! Permitir equipos de la
FW(config-ext-nacl)# permit tcp 193.144.11.0 0.0.0.255 any eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any
```

(5) Denegar el resto de paquetes entrantes

Filtrado Estático de Paquetes

```
! Reglas de entrada en la interfaz externa
FW(config)# ip access-list extended entrada
! Deniega IP-spoofing
FW(config-ext-nacl)# deny ip 193.144.11.0 0.0.0.255 any
! Deniega conexiones al propio firewall
FW(config-ext-nacl)# deny ip any host 193.144.10.10
FW(config-ext-nacl)# deny ip any host 193.144.11.1
! Permite acceso al servidor web
FW(config-ext-nacl)# permit tcp any host 193.144.11.10 eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any

! Reglas de salida en
FW(config-ext-nacl)# ip
! Permitir equipos de la LAN acceder a Internet
FW(config-ext-nacl)# permit tcp 193.144.11.0 0.0.0.255 any eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any
```

(6) Permitir paquetes TCP salientes con origen en la LAN y puerto de destino 80

Filtrado Estático de Paquetes

```
! Reglas de entrada en la interfaz externa
FW(config)# ip access-list extended entrada
! Deniega IP-spoofing
FW(config-ext-nacl)# deny ip 193.144.11.0 0.0.0.255 any
! Deniega conexiones al propio firewall
FW(config-ext-nacl)# deny ip any host 193.144.10.10
FW(config-ext-nacl)# deny ip any host 193.144.11.1
! Permite acceso al servidor web
FW(config-ext-nacl)# permit tcp any host 193.144.11.10 eq 80
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any

! Reglas de salida en la
FW(config-ext-nacl)# ip
! Permitir equipos de la
FW(config-ext-nacl)# perm
! Denegar todo por defecto
FW(config-ext-nacl)# deny ip any any
```

(7) Denegar el resto de paquetes salientes

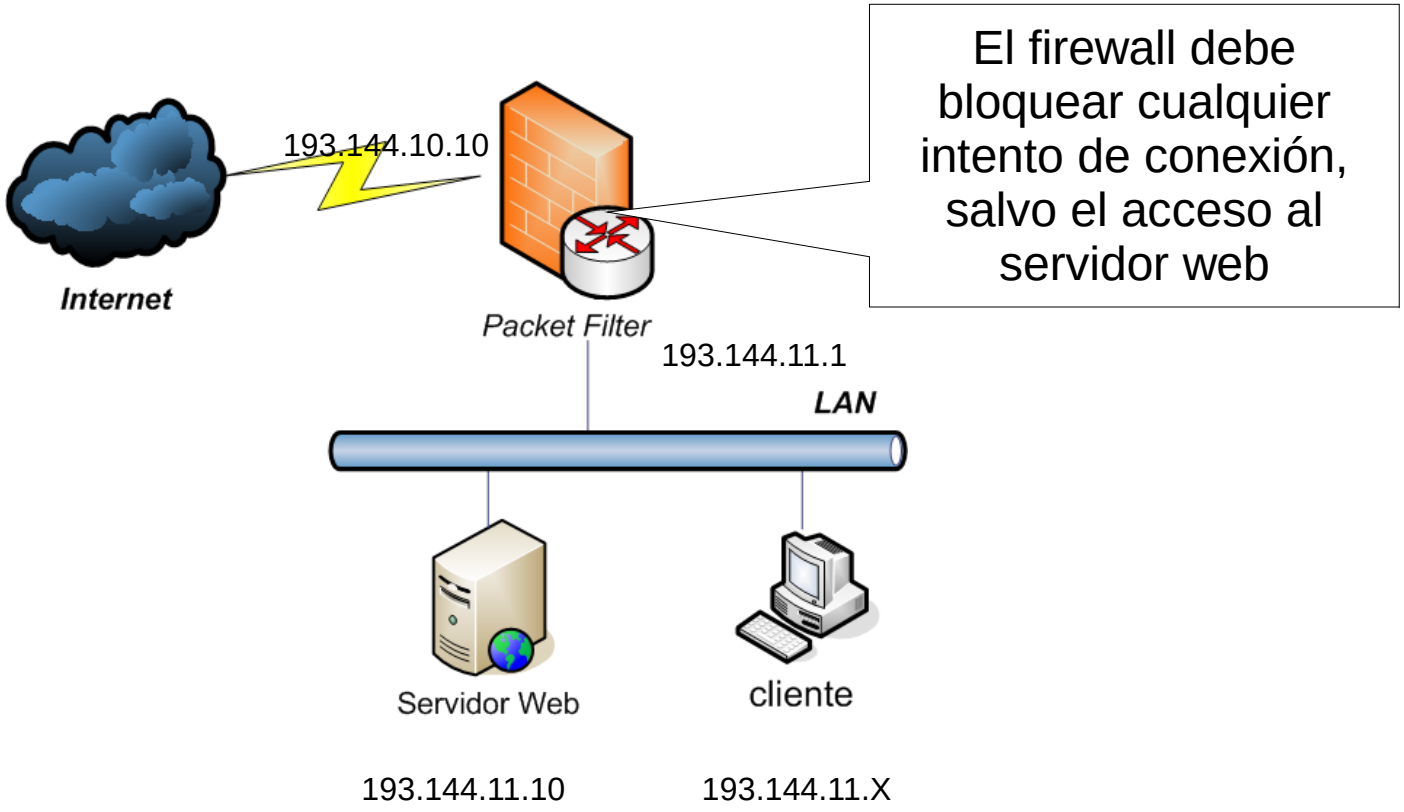
Filtrado Estático de Paquetes

- Conjunto de reglas: ejemplo
  - (1) Denegar paquetes entrantes con IP con origen en la LAN
  - (2) y (3) Denegar paquetes entrantes con destino el propio FW
  - (4) Permitir paquetes TCP entrantes destinados al servidor web y al puerto 80
  - (5) Denegar el resto de paquetes entrantes
  - (6) Permitir paquetes TCP salientes con origen la LAN y destino el puerto 80
  - (7) Denegar el resto de paquetes salientes

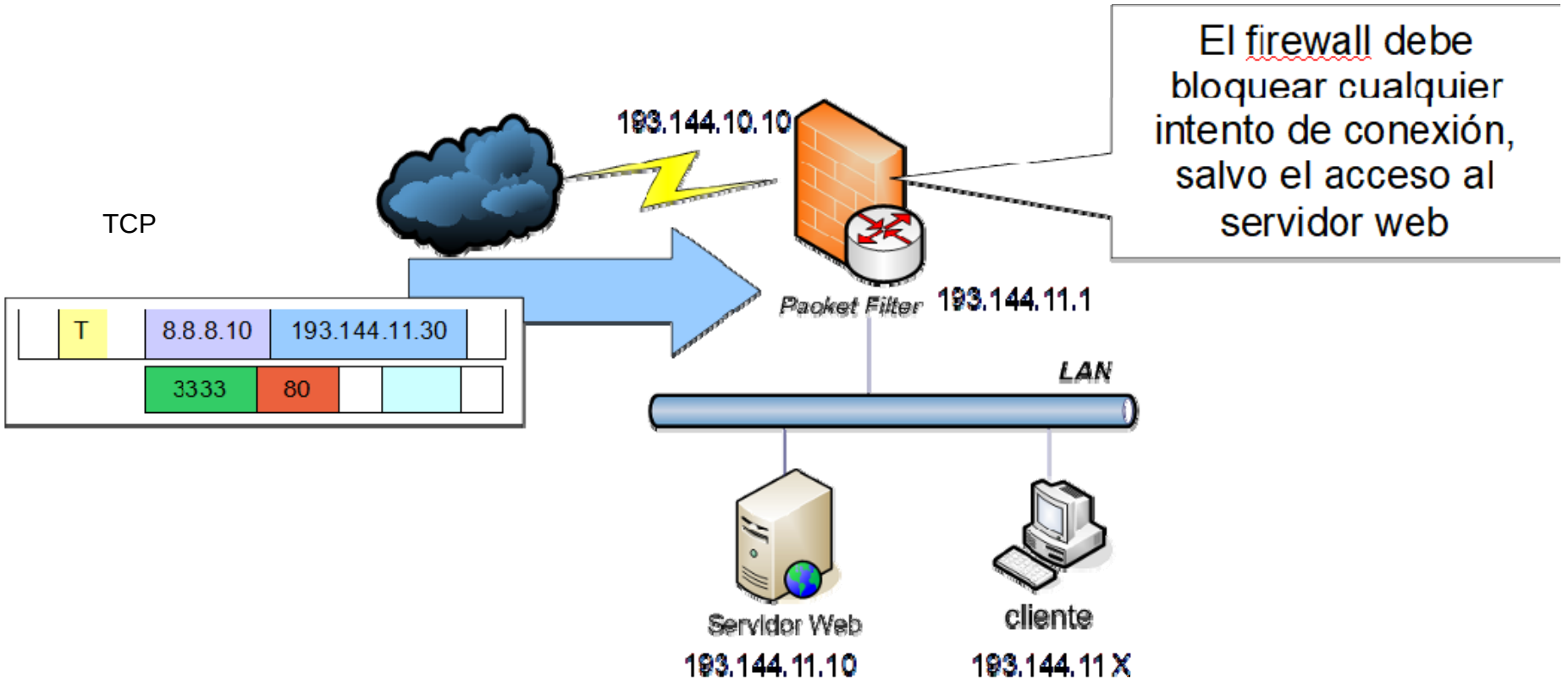
Filtrado estático: Solo mira las cabeceras del paquete (IP, puerto, protocolo) y aplica reglas fijas. No entiende ni sigue la conexión, ni analiza el contenido. Es rápido, simple, pero básico.

Filtrado Estático de Paquetes

- Ejemplo:



Filtrado Estático de Paquetes

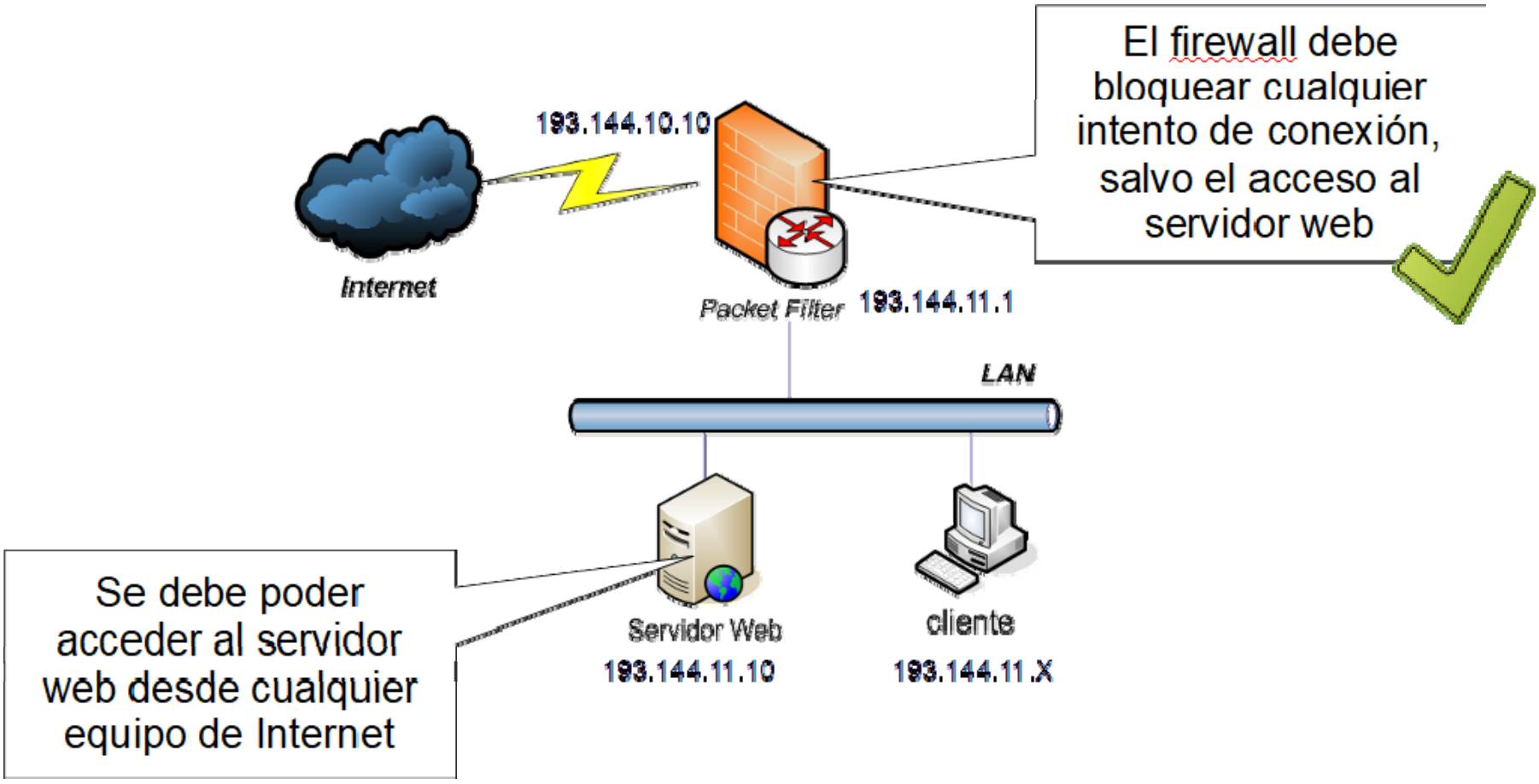


Filtrado Estático de Paquetes

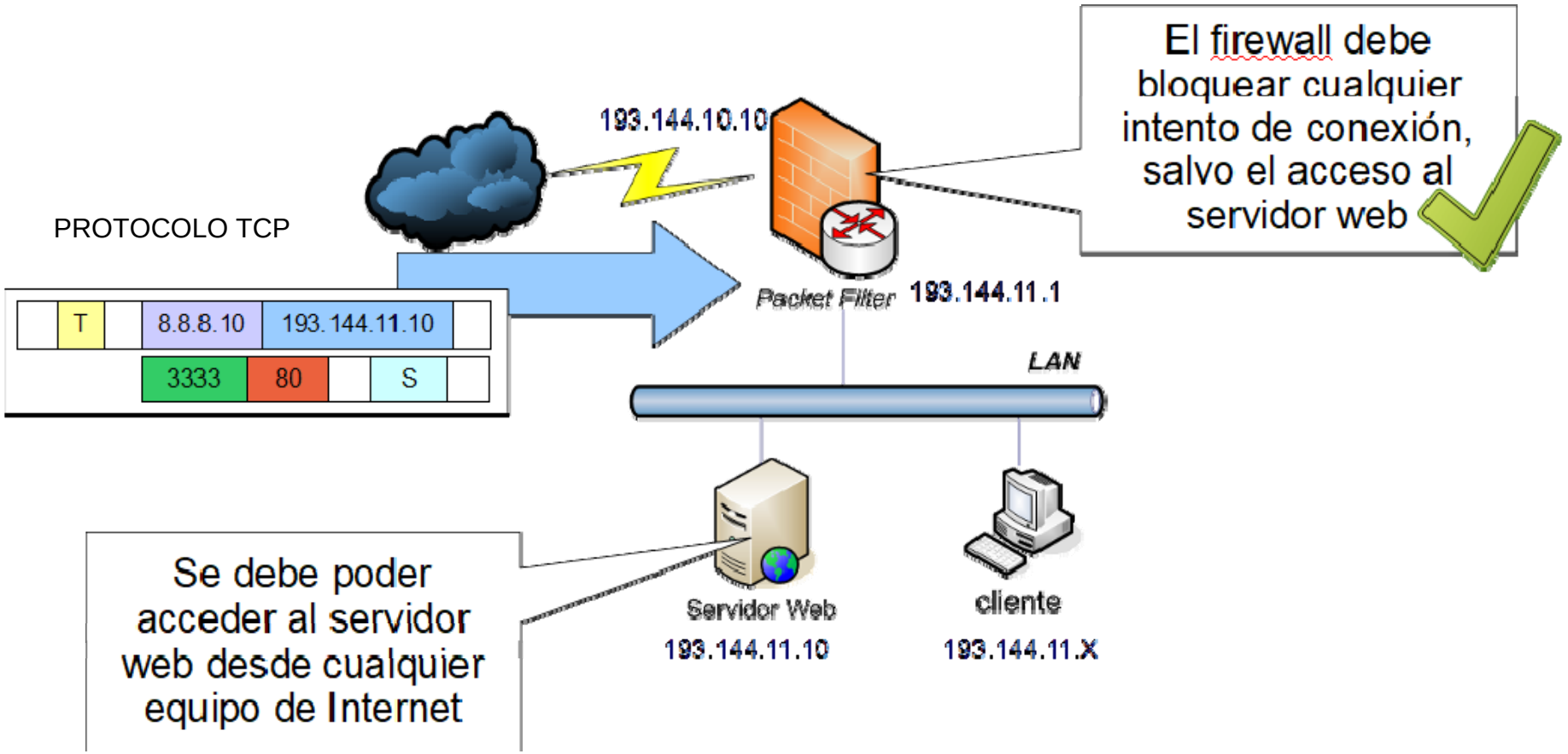
- Conjunto de reglas: ejemplo
  - ✖ (1) Denegar paquetes entrantes con **IP de origen en la LAN** : la ip de origen en la LAN es 8.8.8.10 (PASA)
  - ✖ (2) y (3) Denegar paquetes entrantes con **destino el propio FW** : Destino es 193.144.11.30 (no es firewall), pasa.
  - ✖ (4) Permitir paquetes **TCP** entrantes destinados **al servidor web y al puerto 80** : El paquete es TCP, destino puerto 80 y la IP destino está en la LAN (suponiendo que el servidor web está en esa IP o dentro del rango permitido).
  - ✓ (5) **Denegar** el **resto** de paquetes entrantes
  - (6) Permitir paquetes TCP salientes con origen en la LAN y el puerto 80
  - (7) Denegar el resto de paquetes salientes

LA 4 YA NO LO CUMPLE, ENTONCES VA A LA 5 y se deniega

Filtrado Estático de Paquetes



Filtrado Estático de Paquetes



Se cambia la ip de destino a la del web server
Se añade la flag SYN (indica que el paquete quiere iniciar una conexión TCP)

Filtrado Estático de Paquetes

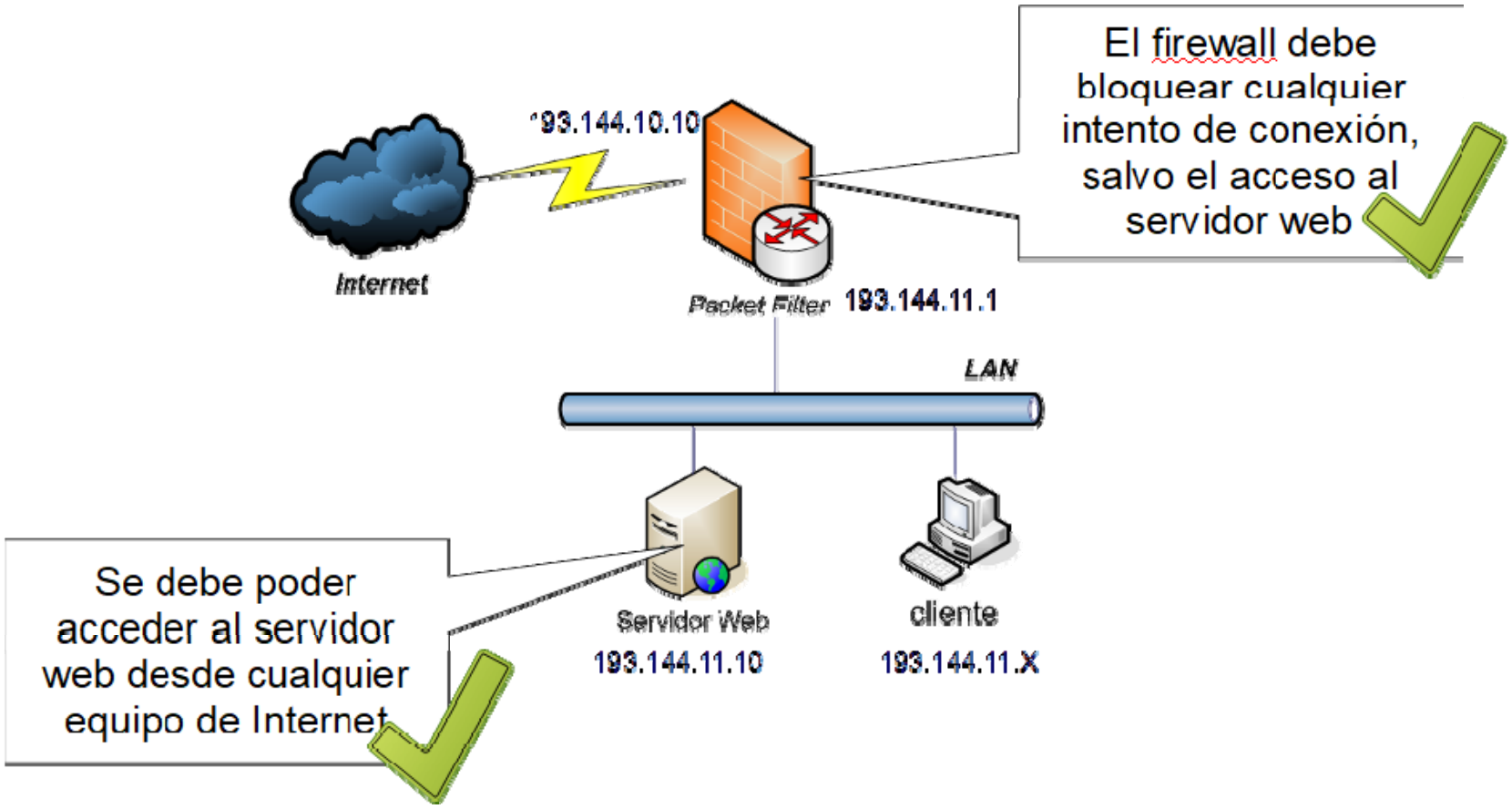
- Conjunto de reglas: ejemplo
  - (1) Denegar paquetes entrantes con **IP de origen en la LAN**
  - (2) y (3) Denegar paquetes entrantes con **destino el propio FW**
  - (4) **Permitir** paquetes **TCP** entrantes destinados **al servidor web y al puerto 80**
  - (5) Denegar el **resto** de paquetes entrantes
  - (6) Permitir paquetes TCP salientes con origen en la LAN y el puerto 80
  - (7) Denegar el resto de paquetes salientes

: Bloquea paquetes con IP de origen de la LAN. no aplica origen es 8.8.8.10

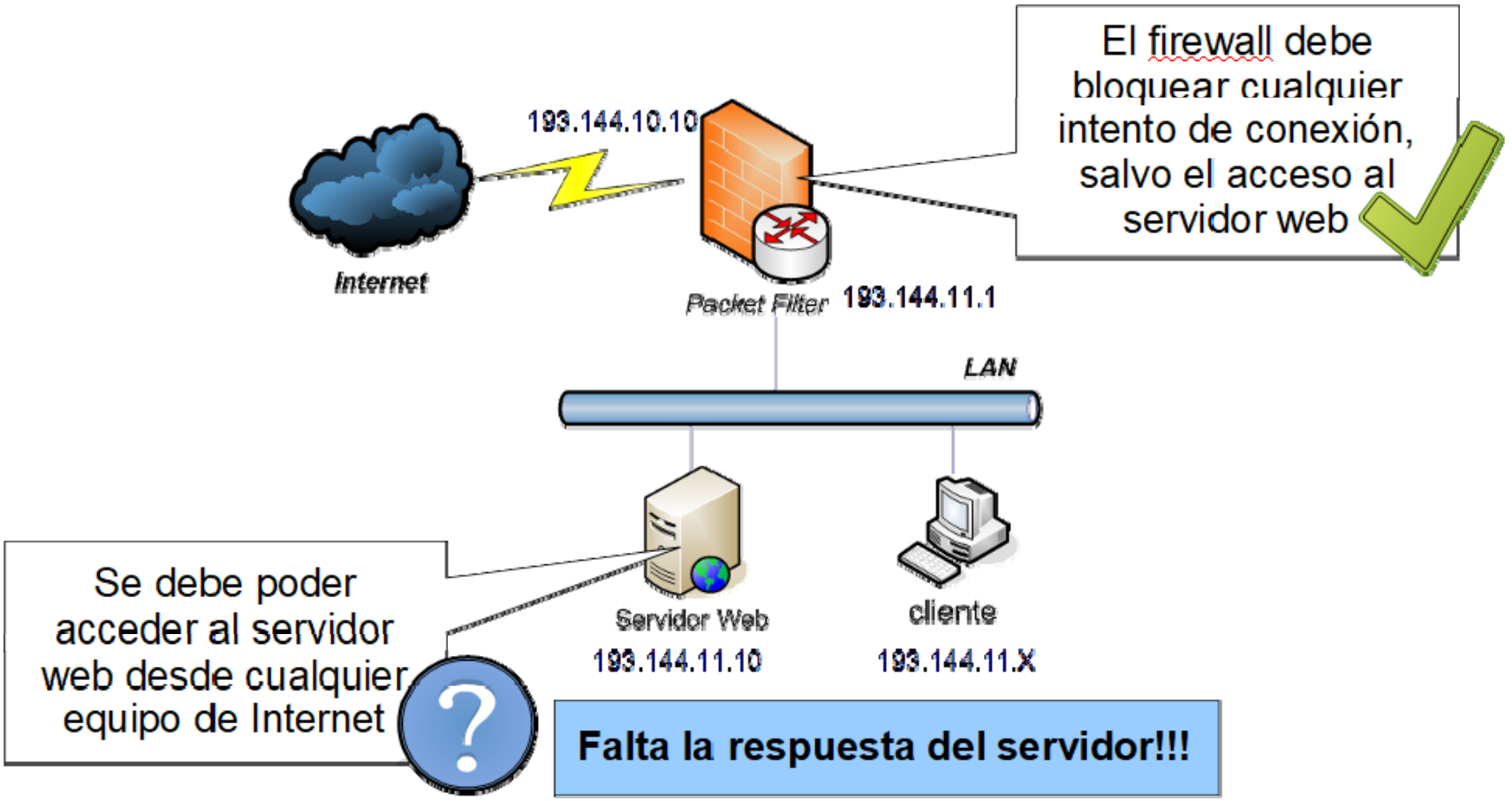
Bloquea paquetes con destino al firewall. (No aplica, destino es 193.144.11.10)

4- Permite tráfico TCP entrante al puerto 80 del servidor web.
(¡Aplica! Coincide IP y puerto, además la flag S indica inicio de conexión web)

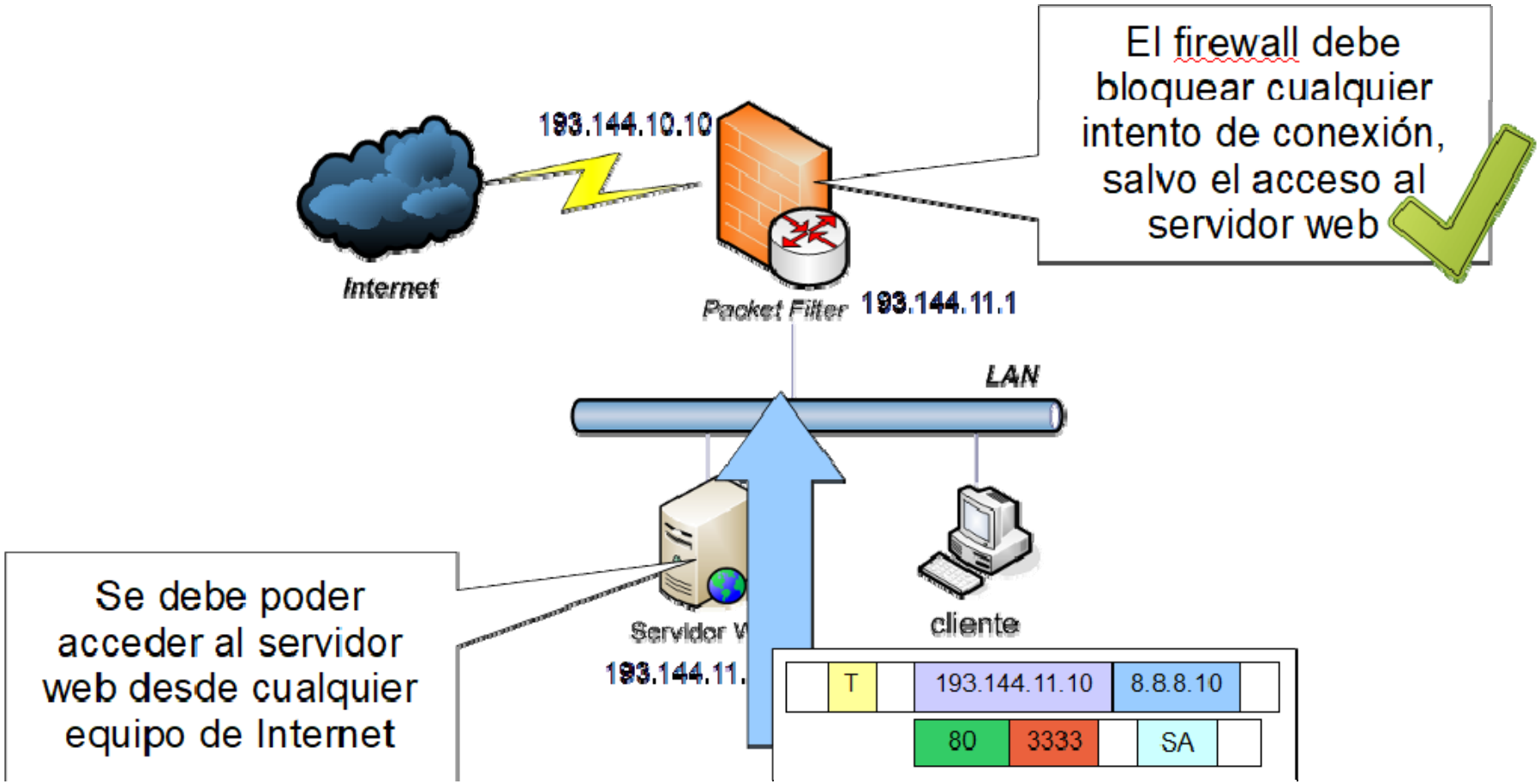
Filtrado Estático de Paquetes



Filtrado Estático de Paquetes



Filtrado Estático de Paquetes

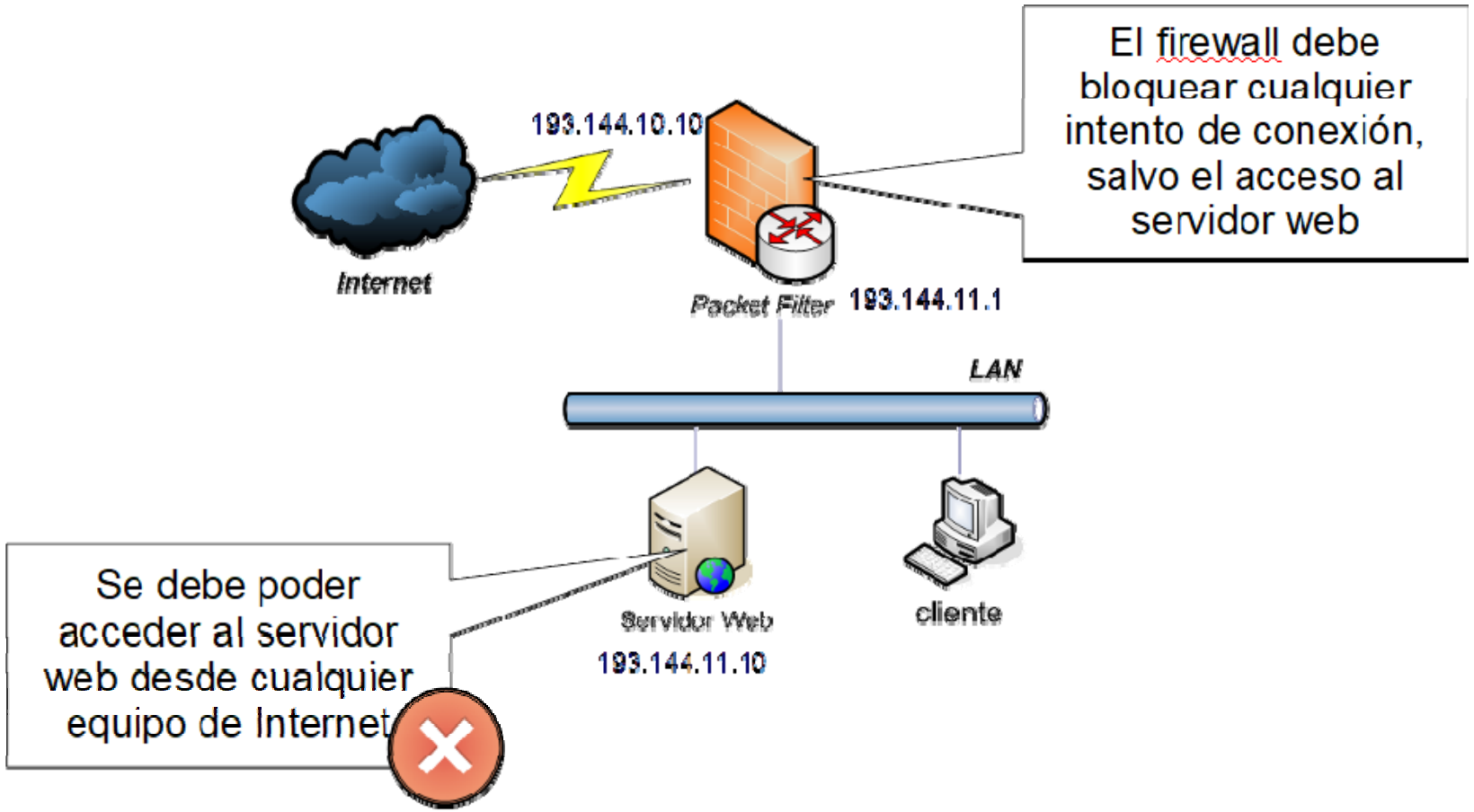


Respuesta del servidor:
No va dejar porque solo deja salir paquetes con puerto de destino 80.
y en este caso el puerto de destino es 3333

Filtrado Estático de Paquetes

- Conjunto de reglas: ejemplo
  - (1) Denegar paquetes entrantes con **IP de origen en la LAN**
  - (2) y (3) Denegar paquetes entrantes con **destino el propio FW**
  - (4) Permitir paquetes **TCP** entrantes destinados **al servidor web y al puerto 80**
  - (5) Denegar el **resto** de paquetes entrantes
  - (6) Permitir paquetes TCP salientes con origen en la LAN y el puerto 80
  - (7) **Denegar** el resto de paquetes salientes

Filtrado Estático de Paquetes



Filtrado Estático de Paquetes

- Ejemplo:
  - ¿Cómo solucionar el problema?

Filtrado Estático de Paquetes

- Ejemplo:
  - ¿Cómo solucionar el problema?
    - Regla para permitir paquetes salientes procedentes del servidor web...

```
permit tcp host 193.144.11.10 any
```

#permite cualquier paquete saliente del servidor

    - ...con puerto de origen 80...

```
permit tcp host 193.144.11.10 eq 80 any
```

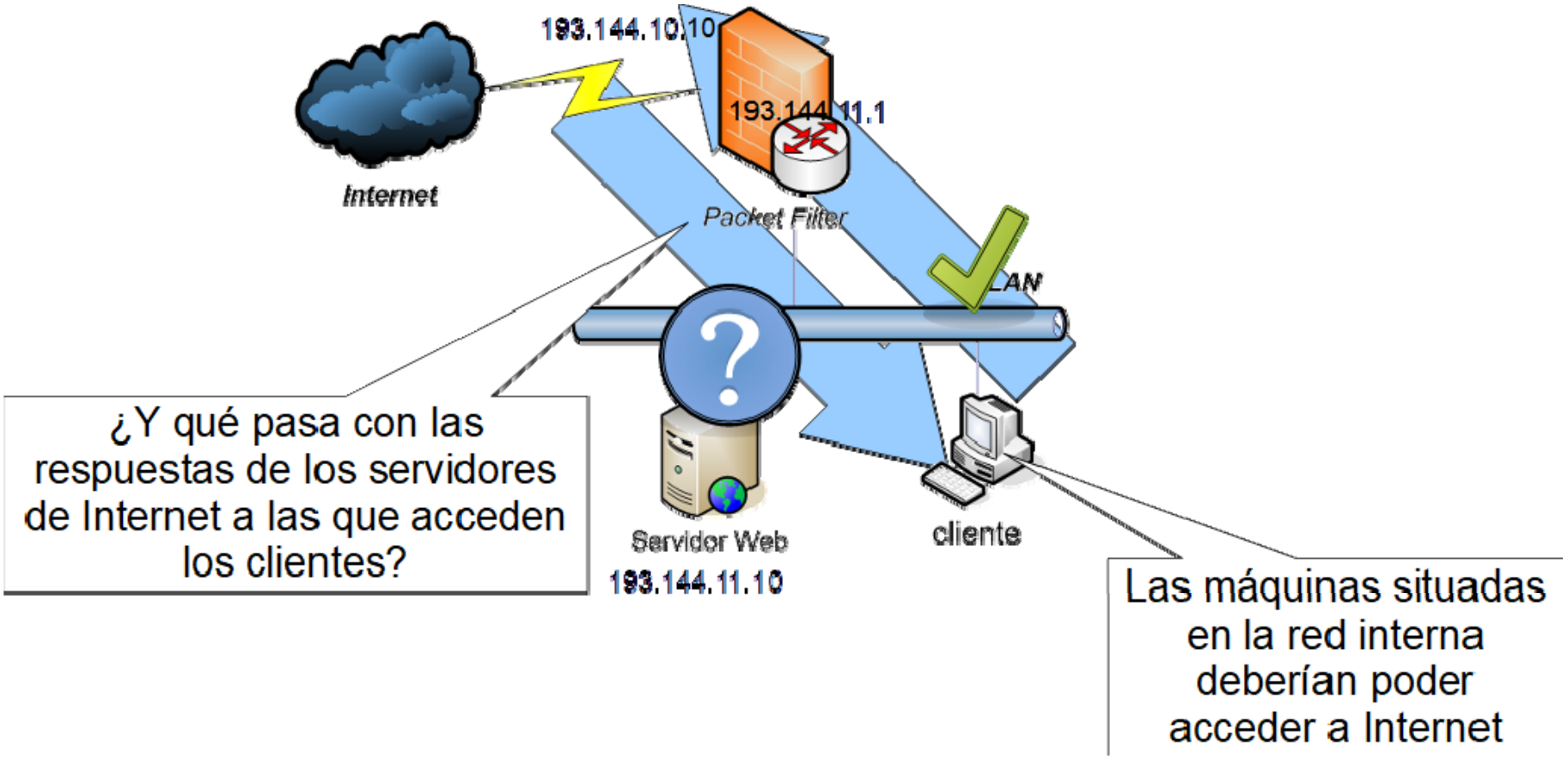
#permite cualquier paquete saliente del servidor en el puerto 80

    - ...y destino por encima de 1023...

```
permit tcp host 193.144.11.10 eq 80 any gt 1023
```

#permite cualquier paquete saliente del servidor en el puerto 80 o cualquier puerto mayor que 1023

Filtrado Estático de Paquetes



Filtrado Estático de Paquetes

- Conjunto de reglas: ejemplo
  - ✖ (1) Denegar paquetes entrantes con **IP de origen en la LAN**
  - ✖ (2) y (3) Denegar paquetes entrantes con **destino el propio FW**
  - ✖ (4) Permitir paquetes **TCP** entrantes destinados **al servidor web y al puerto 80**
  - ✓ (5) **Denegar** el **resto** de paquetes entrantes
  - (6) Permitir paquetes TCP salientes con origen en la LAN y el puerto 80
  - (7) Denegar el resto de paquetes salientes

Solución1: Filtrado dinámico
El firewall recuerda la conexión del cliente y permite automáticamente la respuesta.

Solución2: Añadir una regla como esta (para filtrado estático)
permit tcp any eq 80 193.144.11.0 0.0.0.255

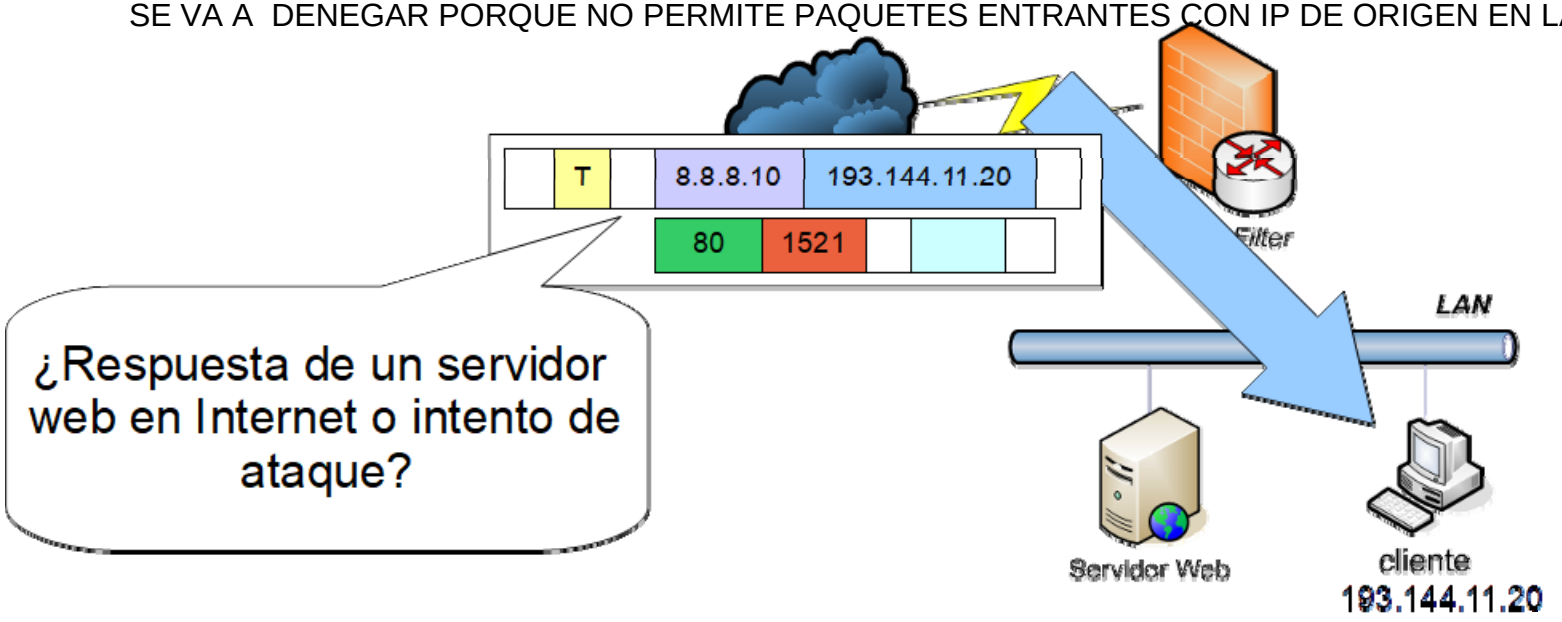
Filtrado Estático de Paquetes

- Ejemplo:
  - ¿Cómo solucionar el problema?
    - Si se utiliza la misma técnica:
      - Regla para permitir paquetes ~~salientes~~ **entrantes** procedentes ~~del servidor web~~ **de Internet**...
      - ...con puerto de origen 80...
      - ...y destino por encima de 1023..

Filtrado Estático de Paquetes

- Ejemplo:
  - Solución no satisfactoria
    - No podemos distinguir una respuesta legítima de un intento de ataque con paquetes de ciertas características

SE VA A DENEGAR PORQUE NO PERMITE PAQUETES ENTRANTES CON IP DE ORIGEN EN LA LAN



Filtrado Estático de Paquetes

- Ejemplo:
  - ¿Podemos hacerlo mejor?
  - Comprobar los flags TCP para distinguir los paquetes que abren una nueva conexión (SYN, !ACK) de otros que no lo hacen (ACK)
    - Seguimos sin poder distinguir entre paquetes pertenecientes a conexiones legítimas y paquetes especialmente manipulados que pueden suponer un ataque (e.g.: escaneo de puertos)
  - ¿Y qué pasa con UDP, ICMP,...?

- Solución: **el firewall debe mantener el estado de cada conexión**

Tecnologías Firewall

2.2- Firewall de Filtrado Dinámico de Paquetes

Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- El firewall no analiza los paquetes individualmente, sino en su contexto (la conexión a la que pertenecen)
  - Añaden conocimiento de **sesión** :El firewall sabe quién inició la conexión y si sigue activa, y decide en base a eso.
  - Filtran conexiones, no paquetes : Si la conexión fue permitida, todos sus paquetes pasan.

Si ya le dejo pasar a un paquete, la respuesta le deja pasar de vuelta si o si:

El firewall mantiene una TABLA DE PAQUETES SALIENTES y permite así las respuestas de los paquetes entrantes en base a conexionex existentes previamente autorizadas como:

- Conexiones TCP
- Respuestas a paquetes UDP previamente enviados (e.g.: DNS)
- Errores ICMP (e.g.: destination unreachable).
- Respuestas ICMP (e.g: echo reply)

Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

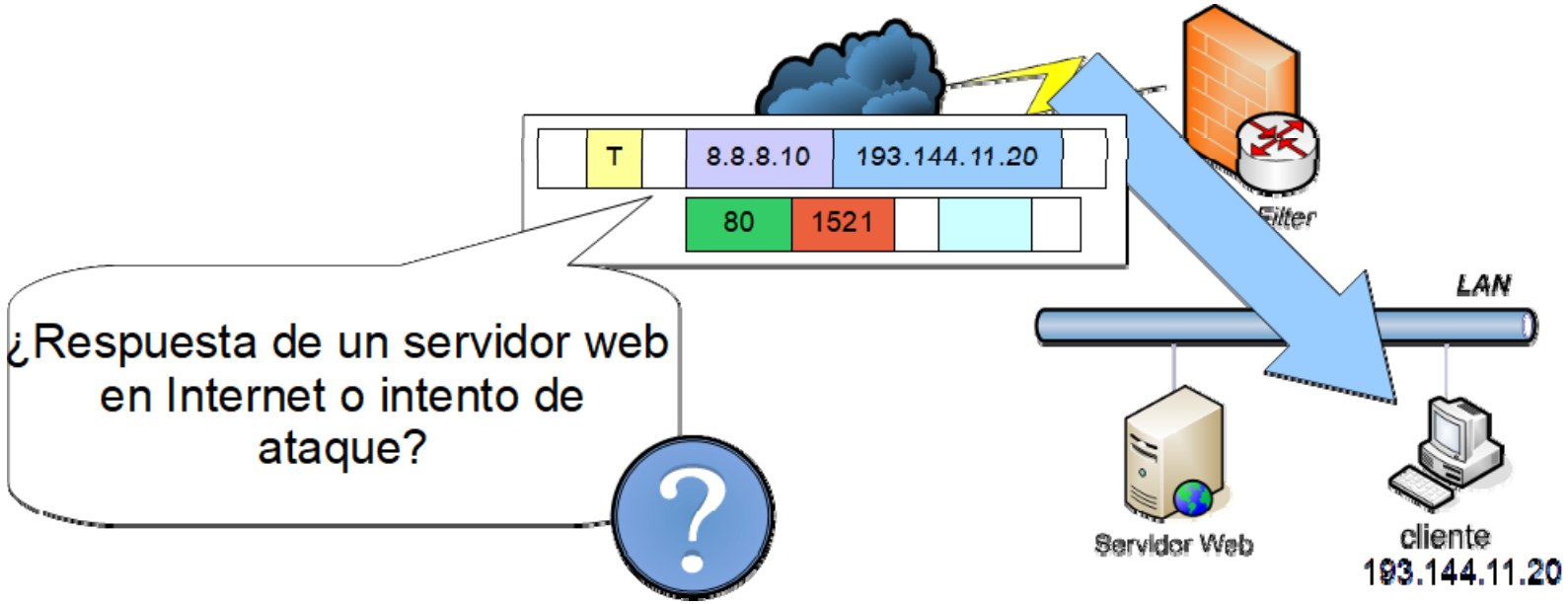
- Operativa:
  - Mantienen una tabla con el estado de las conexiones salientes, que utilizan para validar los paquetes entrantes
    - Se adapta bien a la operativa de TCP
    - También mantiene entradas para UDP (puerto + IP), ICMP,… y en general otros protocolos transportados en IP (aunque la gestión de estas sesiones es menos precisa)
  - Mayor carga
  - Permiten paquetes de entrada sólo si están relacionados con una conexión existente previamente autorizada.
    - Conexiones TCP
    - Respuestas a paquetes UDP previamente enviados (e.g.: DNS)
    - Errores ICMP (e.g.: *destination unreachable*).
    - Respuestas ICMP (e.g: *echo reply*)

Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- Ejemplo
  - (1) Permitir conexiones HTTP desde Internet al servidor web
  - (2) Permitir conexiones de los equipos de la LAN a Internet.
  - (3) Denegar el resto

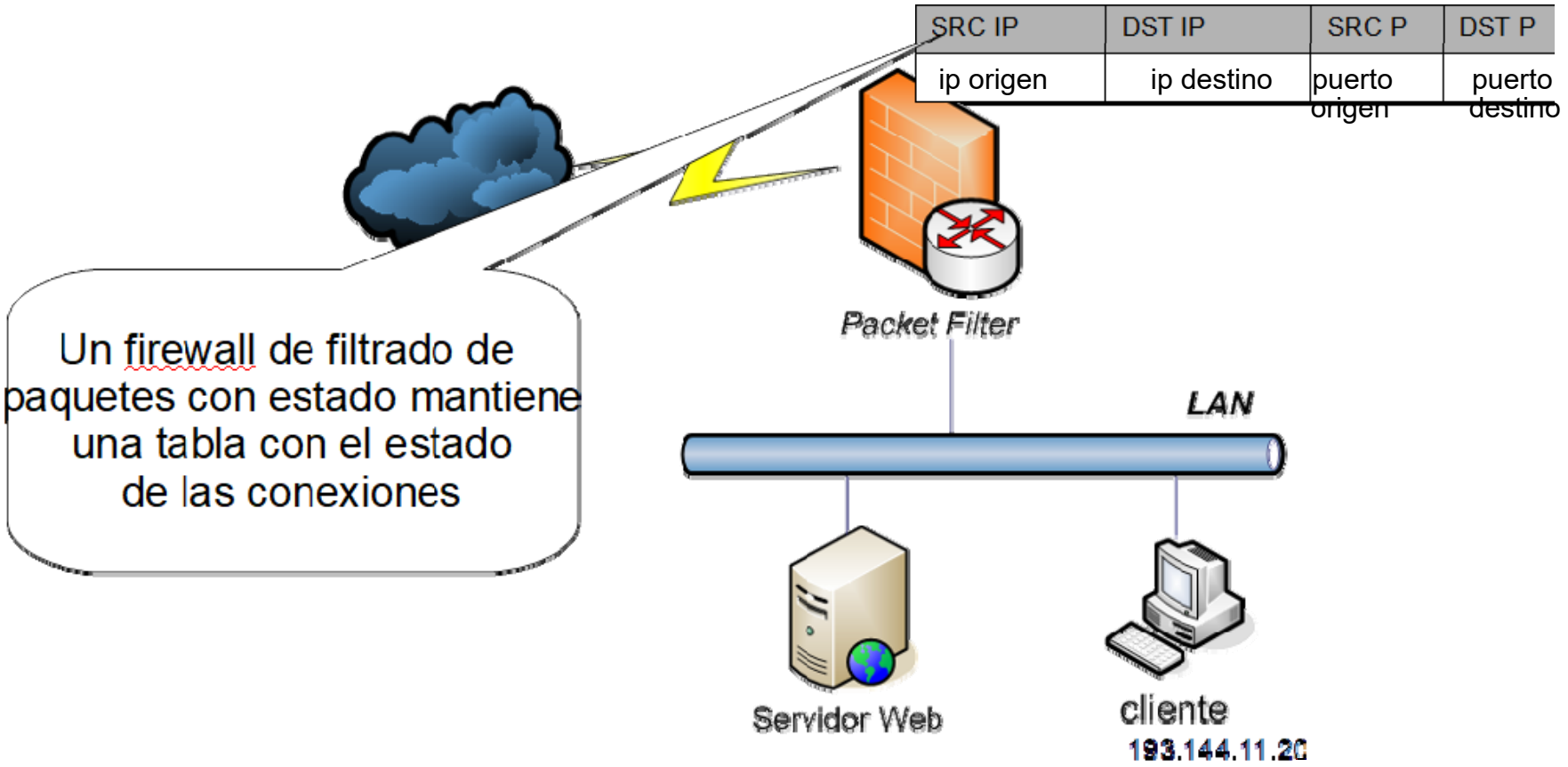
Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- Estado de las conexiones
  - En un firewall de filtrado de paquetes estático, tenemos el problema de manejar adecuadamente las respuestas. E.g.:



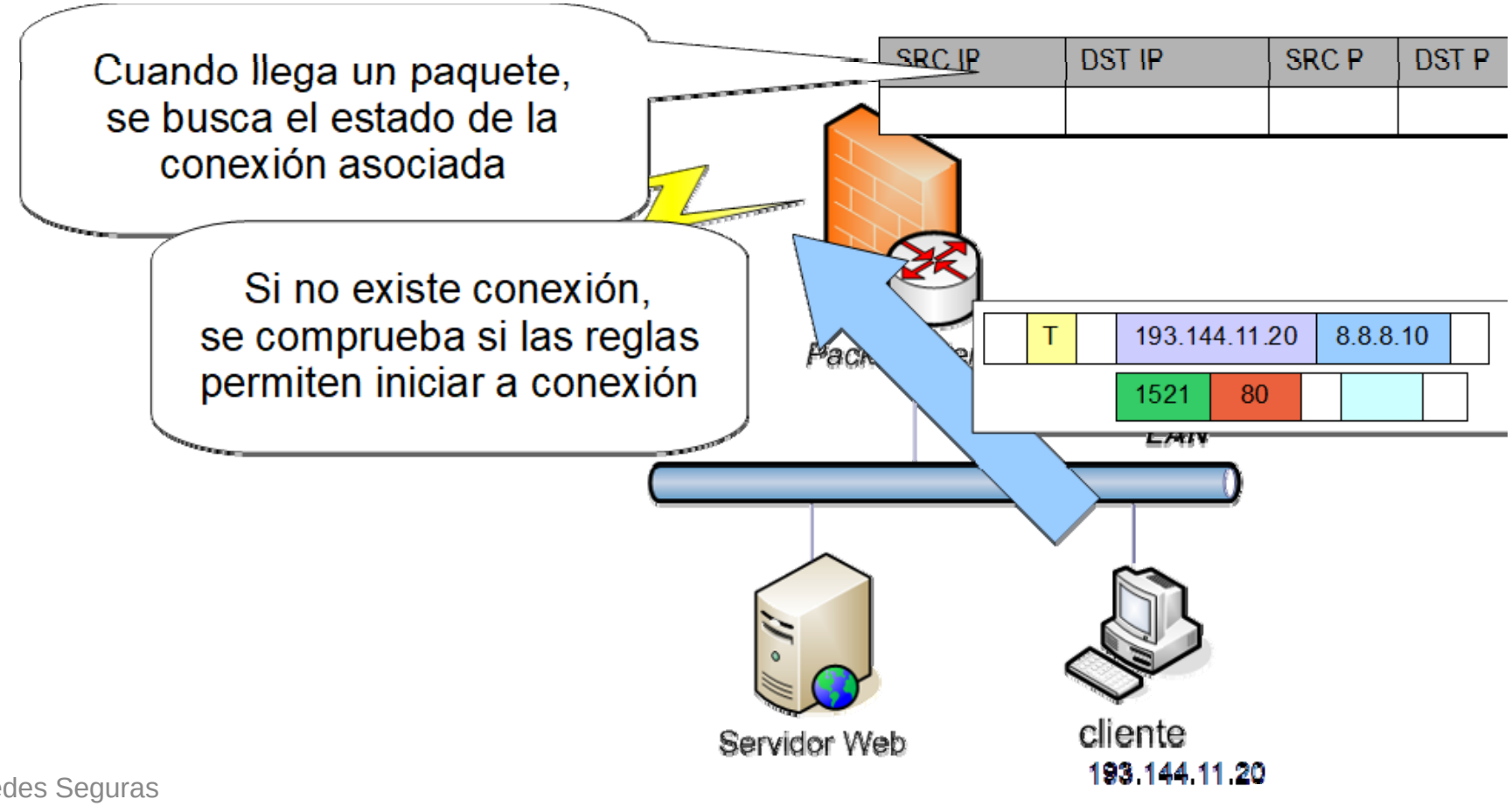
Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- Estado de las conexiones



Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- Estado de las conexiones



Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- Ejemplo:

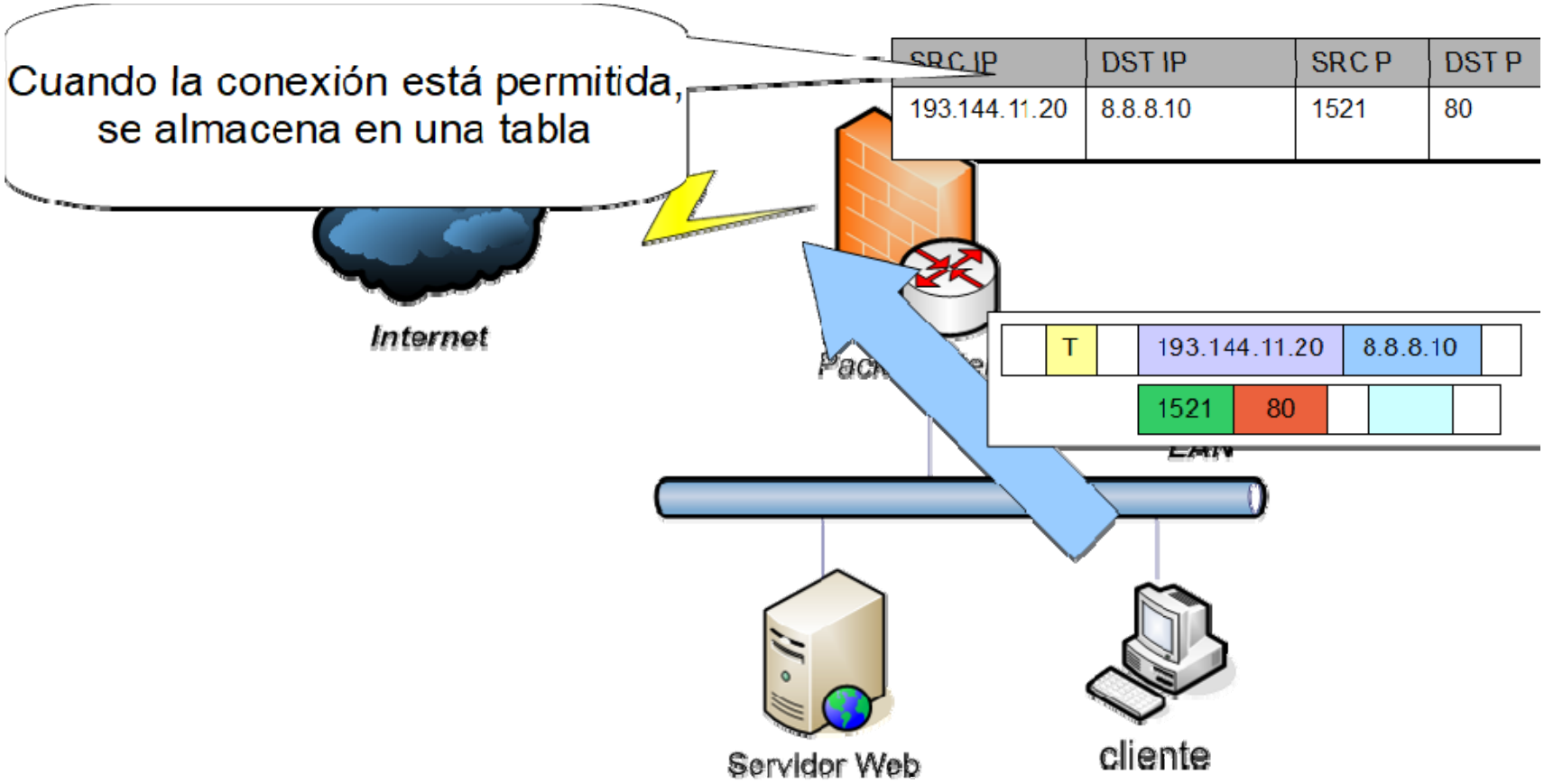

(1) Permitir conexiones HTTP desde Internet al servidor web


(2) **Permitir** conexiones de los equipos de la LAN a Internet.

(3) Denegar o resto

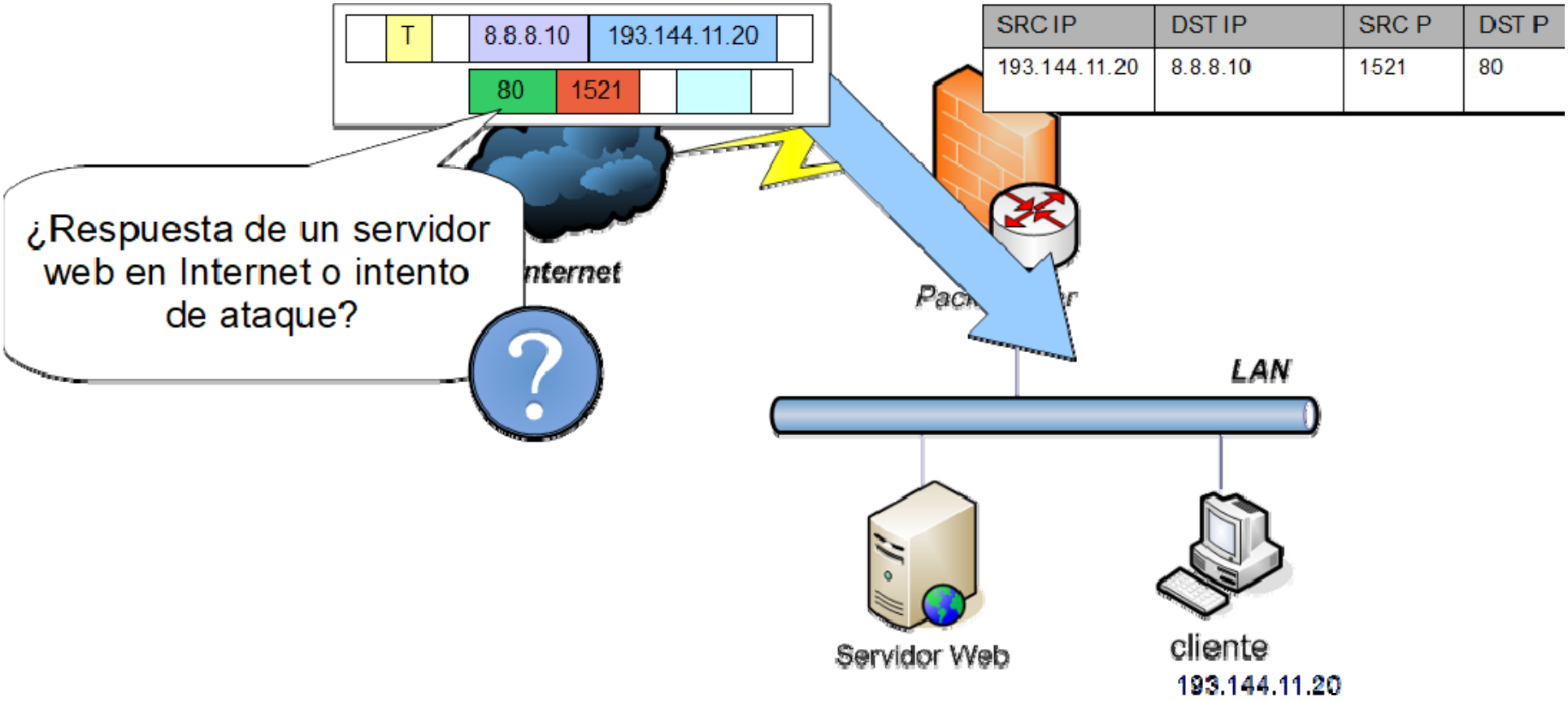
Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- Estado de las conexiones



Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- Estado de las conexiones:



Filtrado Dinámico de Paquetes o *Stateful Packet Filtering*

- Estado de las conexiones

Conclusiones sobre los Firewalls de Filtrado de Paquetes

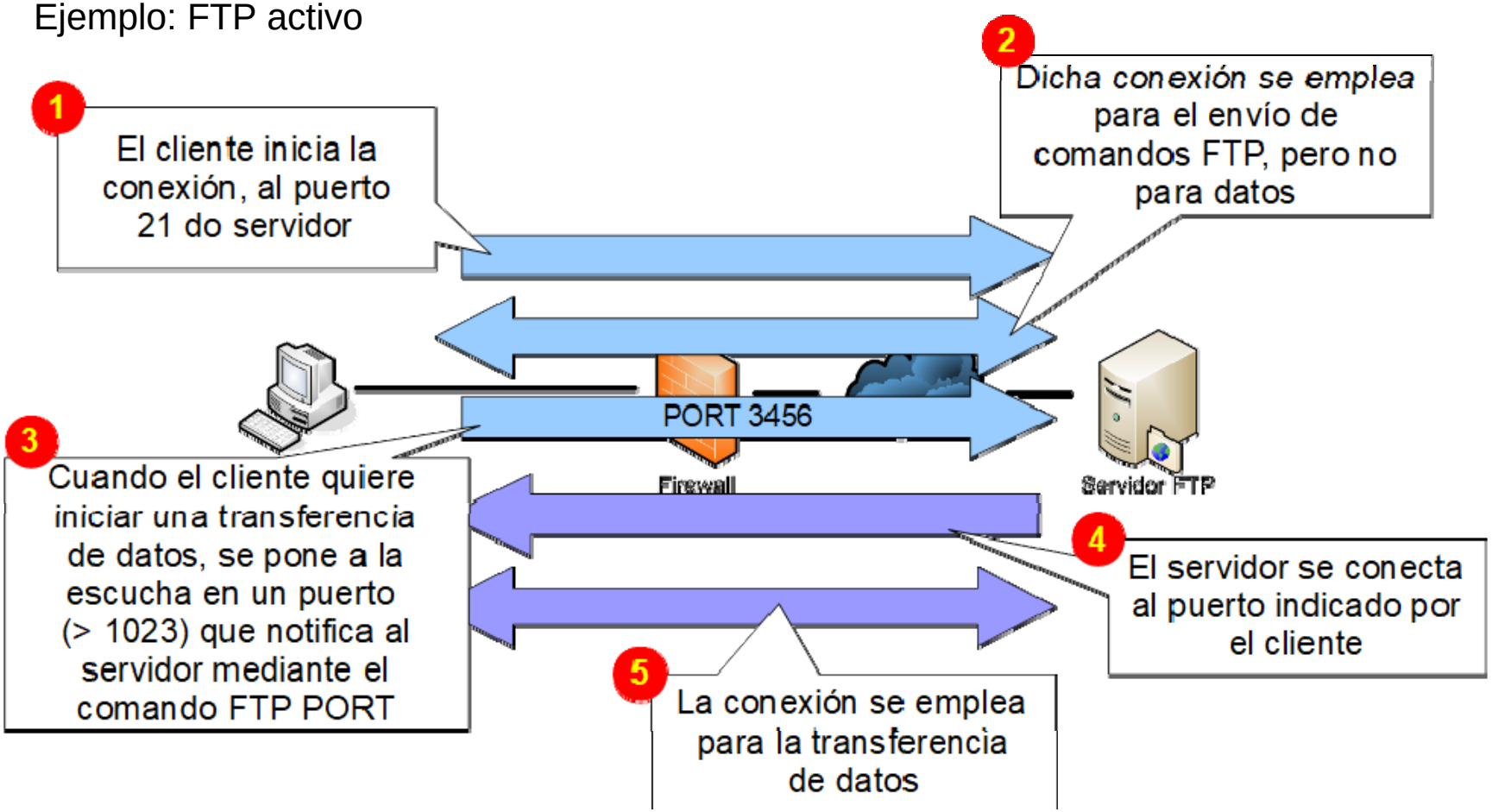
- Características
  - Permiten el control de la mayor parte de protocolos empleados actualmente
  - Eficiencia: solamente examinan ciertos campos del paquete
    - El filtrado de paquetes con estado es más costoso
- Utilidad
  - Control de tráfico en redes
    - Bloqueo/Autorización de ciertos servicios
    - Reducción de la carga en la red interna
- Protección frente a múltiples ataques que aprovechan vulnerabilidades de TCP/IP (e.g.: ciertos casos de IP-Spoofing, ataques de denegación de servicio basados en ICMP...)

Buenas Prácticas en Firewalls de Filtrado de Paquetes

- Se recomienda bloquear:
  - Tráfico hacia el propio firewall desde orígenes no autenticados, para evitar ataques contra el propio firewall
  - Tráfico ICMP, salvo excepciones controladas para evitar (DoS, mapeo de rede...)
  - Tráfico con direcciones no válidas:
    - Tráfico entrante con una IP de origen perteneciente a una red tras el firewall (intento de *spoofing*)
    - Tráfico entrante con dirección de origen privada (RFC 1918)
    - Tráfico de entrada o salda con origen/destino 127.0.0.1 (ataque contra o firewall)
    - Paquetes con direcciones IP de origen “marcianas”: *broadcast*, *multicast* o 0.0.0.0
  - Cierto tipo de paquetes IP: opción *IP Source Routing*, paquetes fragmentados (o reensamblar), ...
  - En general, todo el tráfico salvo aquel necesario (política restrictiva)

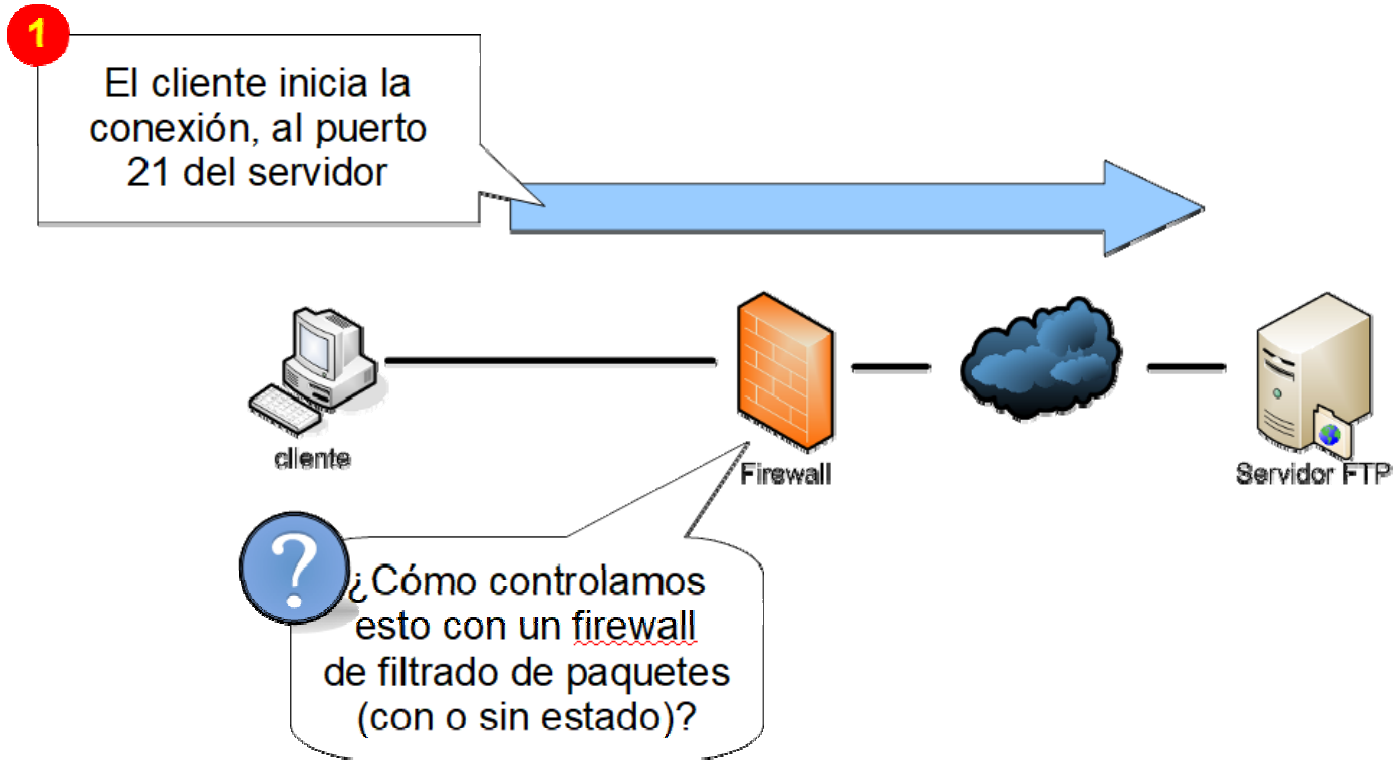
Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: FTP activo



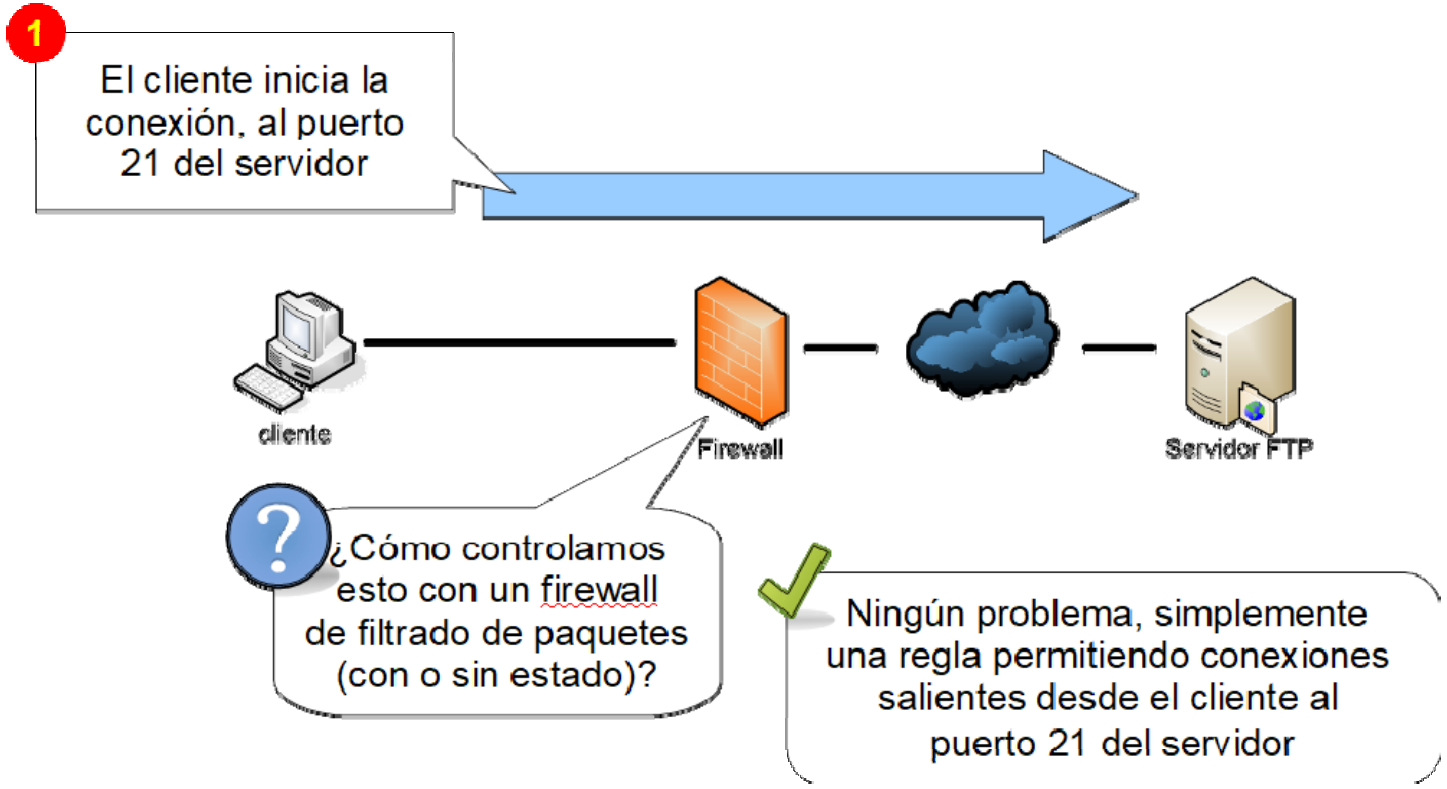
Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: FTP activo



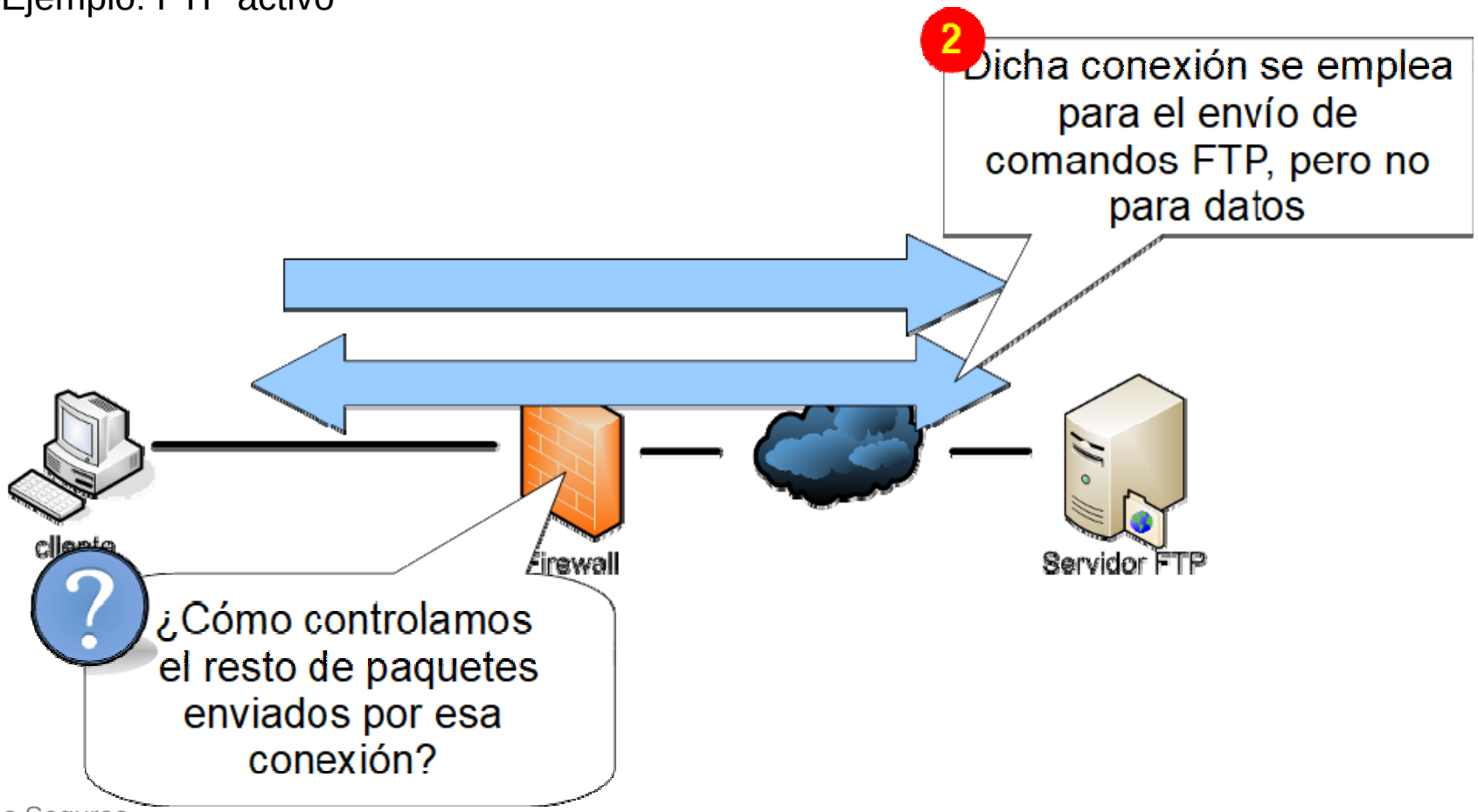
Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: FTP activo



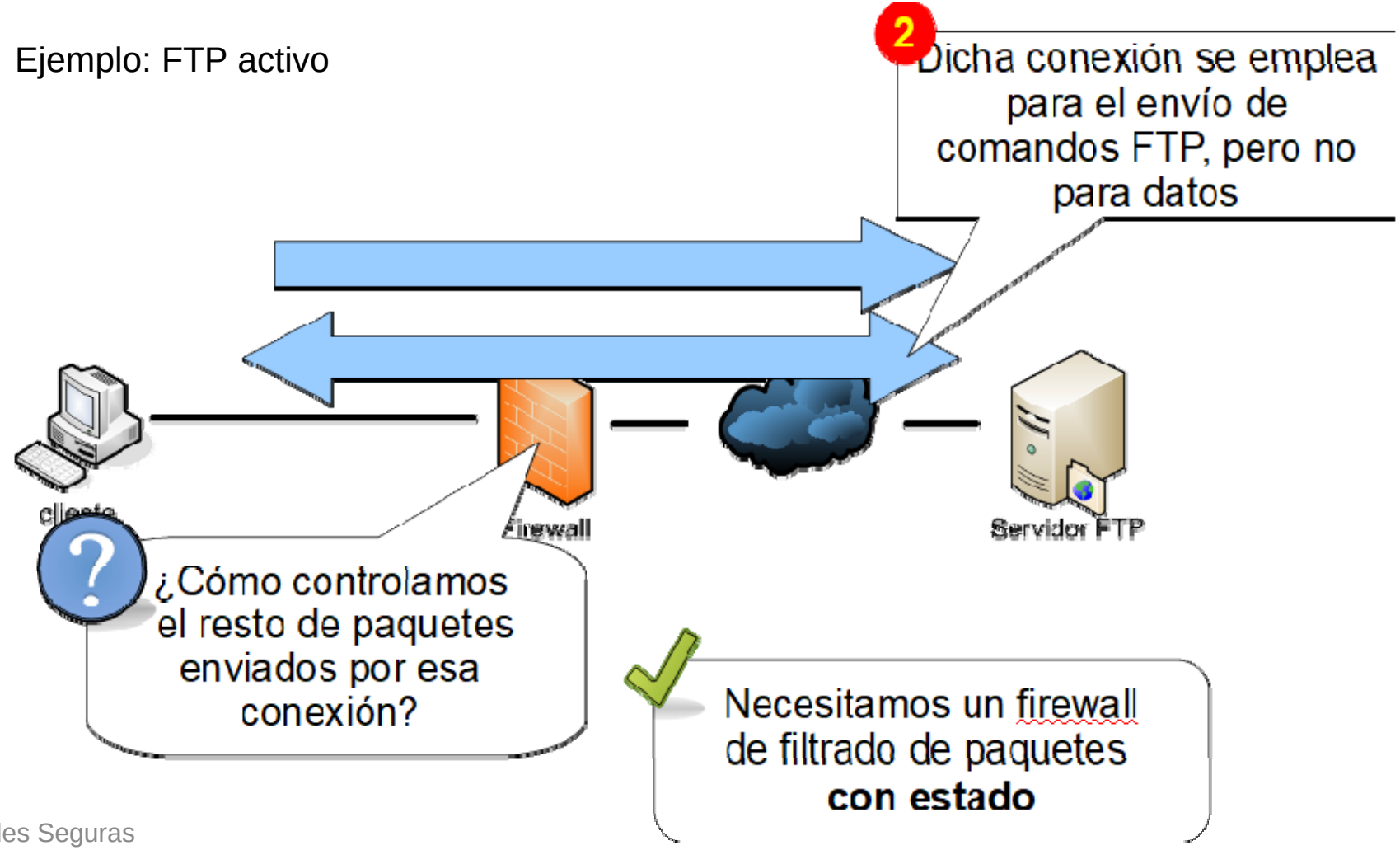
Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: FTP activo



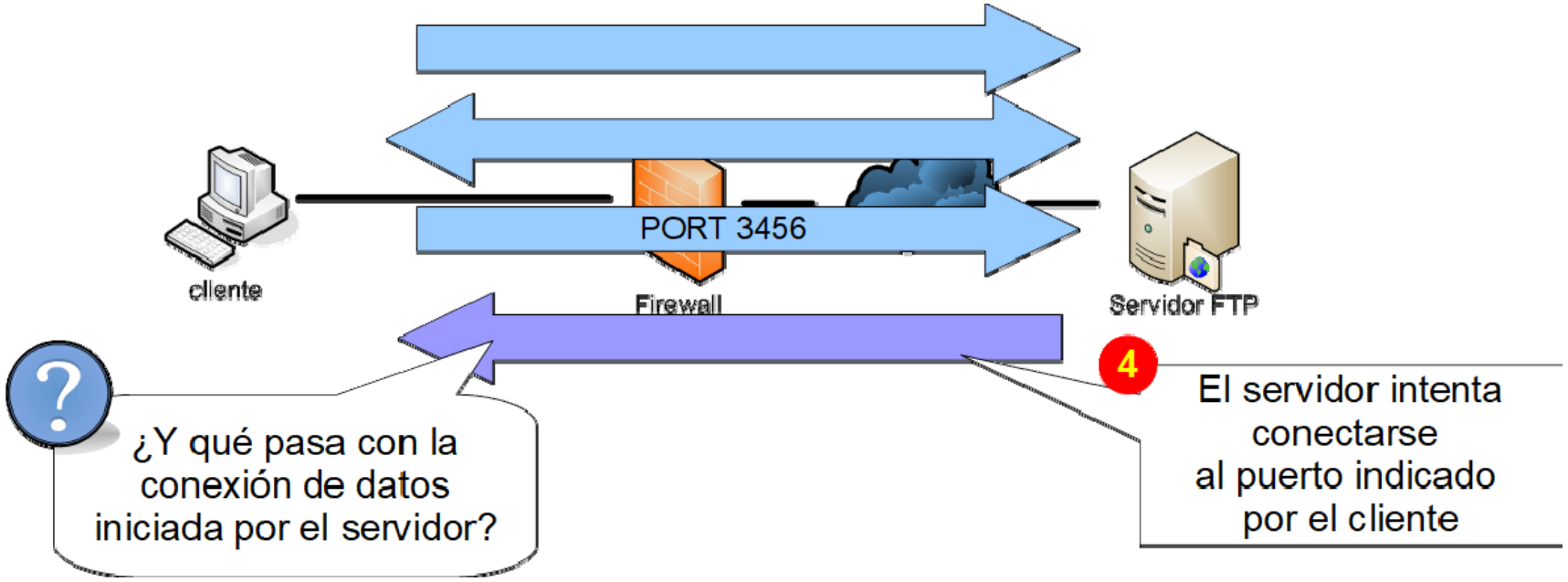
Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: FTP activo



Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo FTP activo:



Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: FTP activo

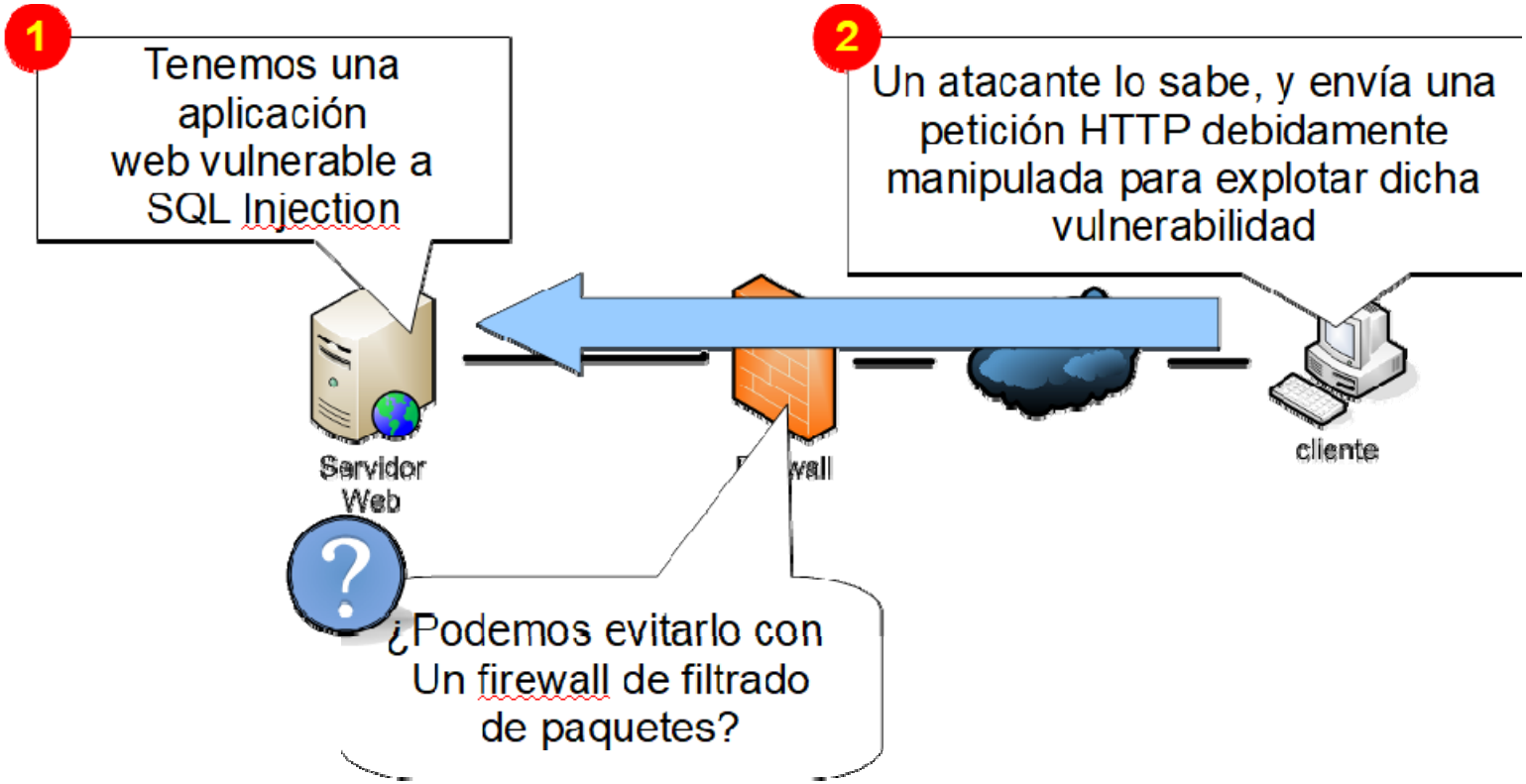
  -> algunos protocolos como FTP son un problema ya que, el firewall solo ve direcciones y puertos, pero en FTP cada vez que se establece una conexión, la respuesta cambia su puerto

  - No conocemos el puerto en el cliente
    - El firewall no sabe interpretar el comando PORT (es un comando a nivel de aplicación)
  - ¿Permitir toda conexión entrante?
    - Problema de seguridad
    - No podemos distinguir conexiones legítimas de ataques
  - La conexión siempre tiene como origen el puerto 20 del servidor
    - Sería trivial para un atacante usar ese porto como origen

- Conclusión:
  - Limitaciones debido a problemas para gestionar cierto tipo de protocolos (e.g.: FTP activo, VoIP, etc.)

Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: SQL Injection

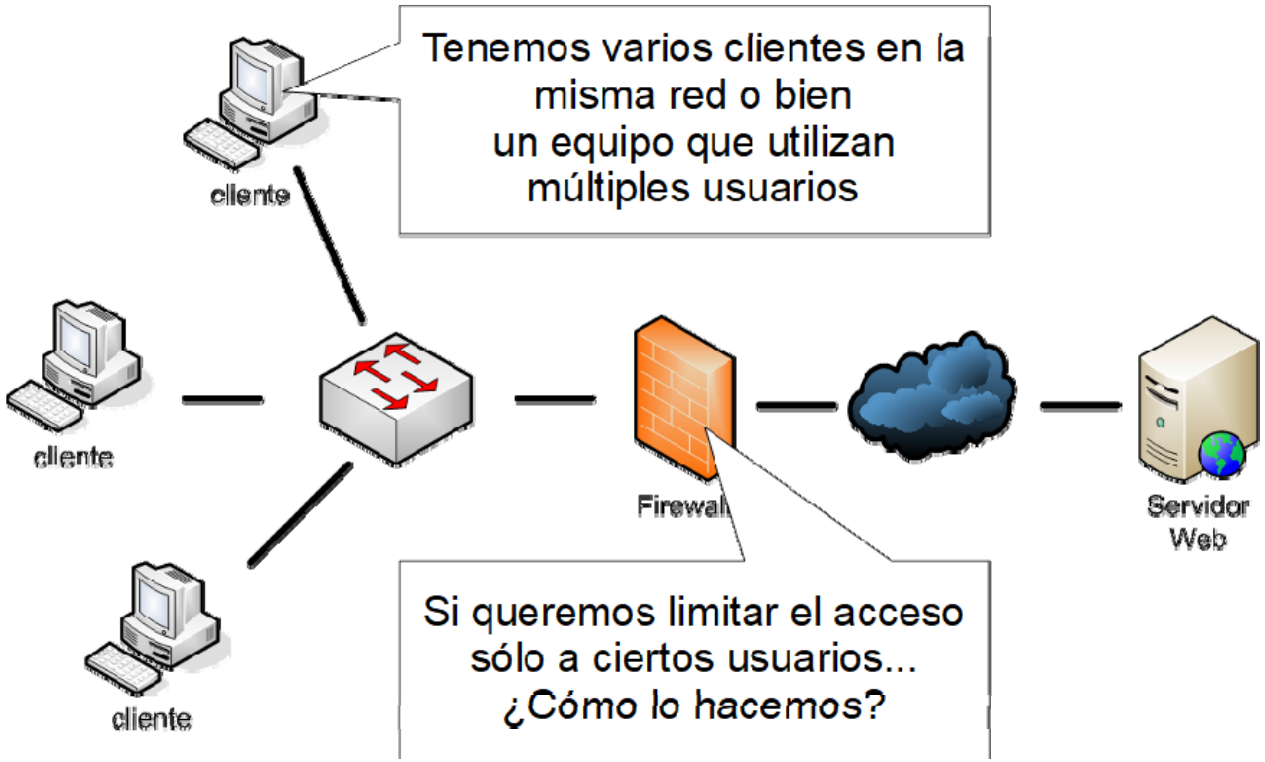


Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: SQL Injection
  - Podemos bloquear las conexiones HTTP al servidor web (bloqueando el puerto 80)
    - Evitamos el ataque
    - Pero también los accesos legítimos
  - No podemos distinguir entre petición legítimas e intentos de ataque
    - El ataque es de nivel de aplicación: la petición HTTP está creada para explotar la vulnerabilidad, pero a nivel TCP e IP la conexión es legítima
  - ¿Se puede conocer la identidad del atacante en base a la información de *log* del firewall?
    - IP, puerto… pero no la identidad del usuario

Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: restricciones de acceso en base al usuario



Limitaciones en Firewalls de Filtrado de Paquetes

- Ejemplo: acceso por usuario
  - ¿Por IP?
    - *IP Spoofing* trivial dentro de la misma subred
    - No es una solución en equipos compartidos por varios usuarios
  - No se puede resolver con firewalls de filtrado de paquetes

Limitaciones de los Firewall de Filtrado de Paquetes

- Problemas para gestionar cierto tipo de protocolos (e.g.: FTP activo)
  - Solución: Firewalls de capa de aplicación, Firewalls de nueva generación
- No impiden ataques que aprovechan vulnerabilidades a nivel de aplicación
  - Solución: Proxies, Firewalls de capa de aplicación, Firewalls de nueva generación e IPS
- Capacidad de *logging* limitada: IP, puerto,...
  - Solución: Proxy, IEEE 802.1x
- No soportan autenticación de usuarios (control de acceso sólo por IP)
  - Solución: Proxy, IEEE 802.1x

Tecnologías Firewall

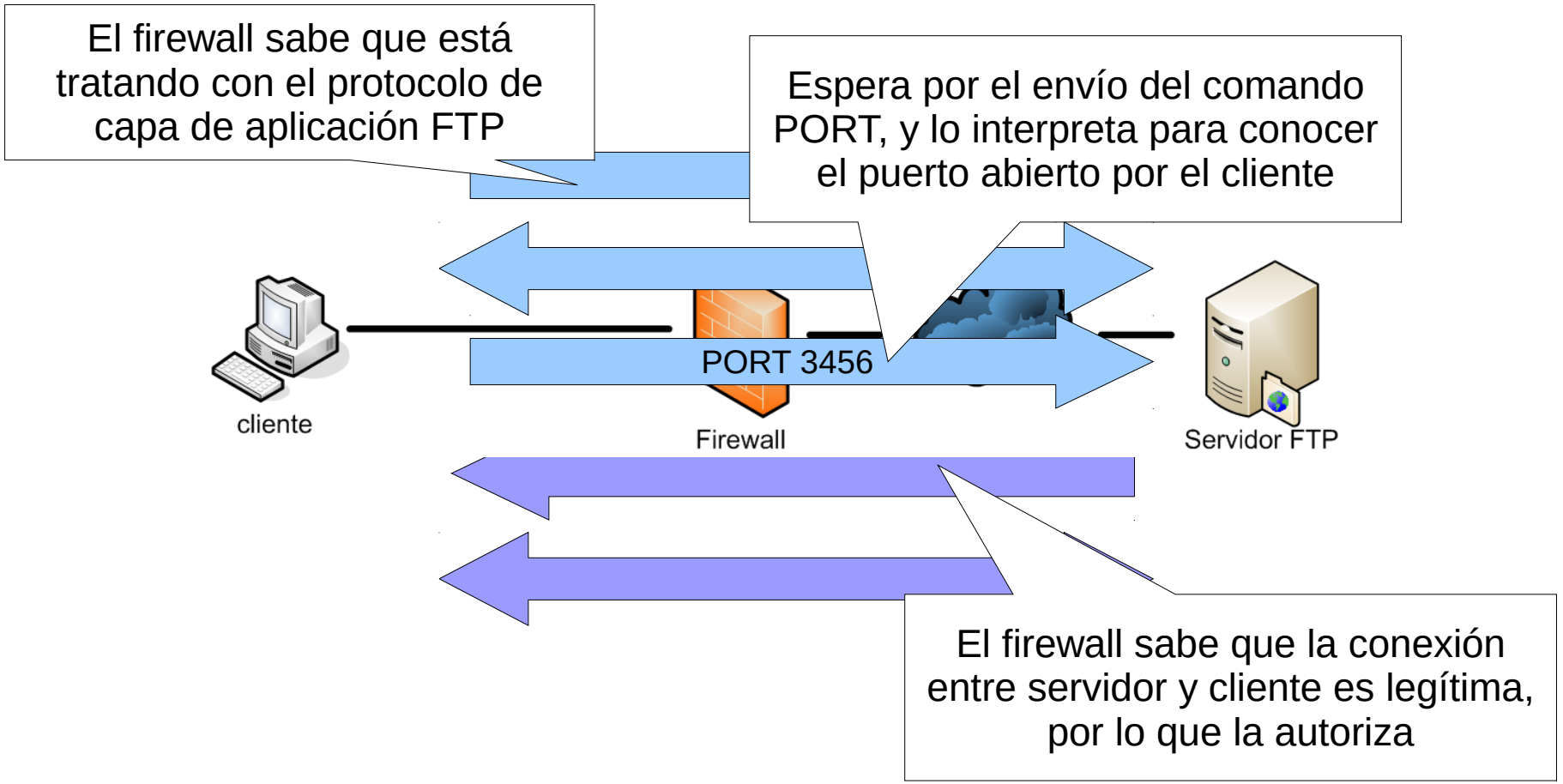
2.3- Firewall de Capa de Aplicación

Filtrado de Capa de Aplicación

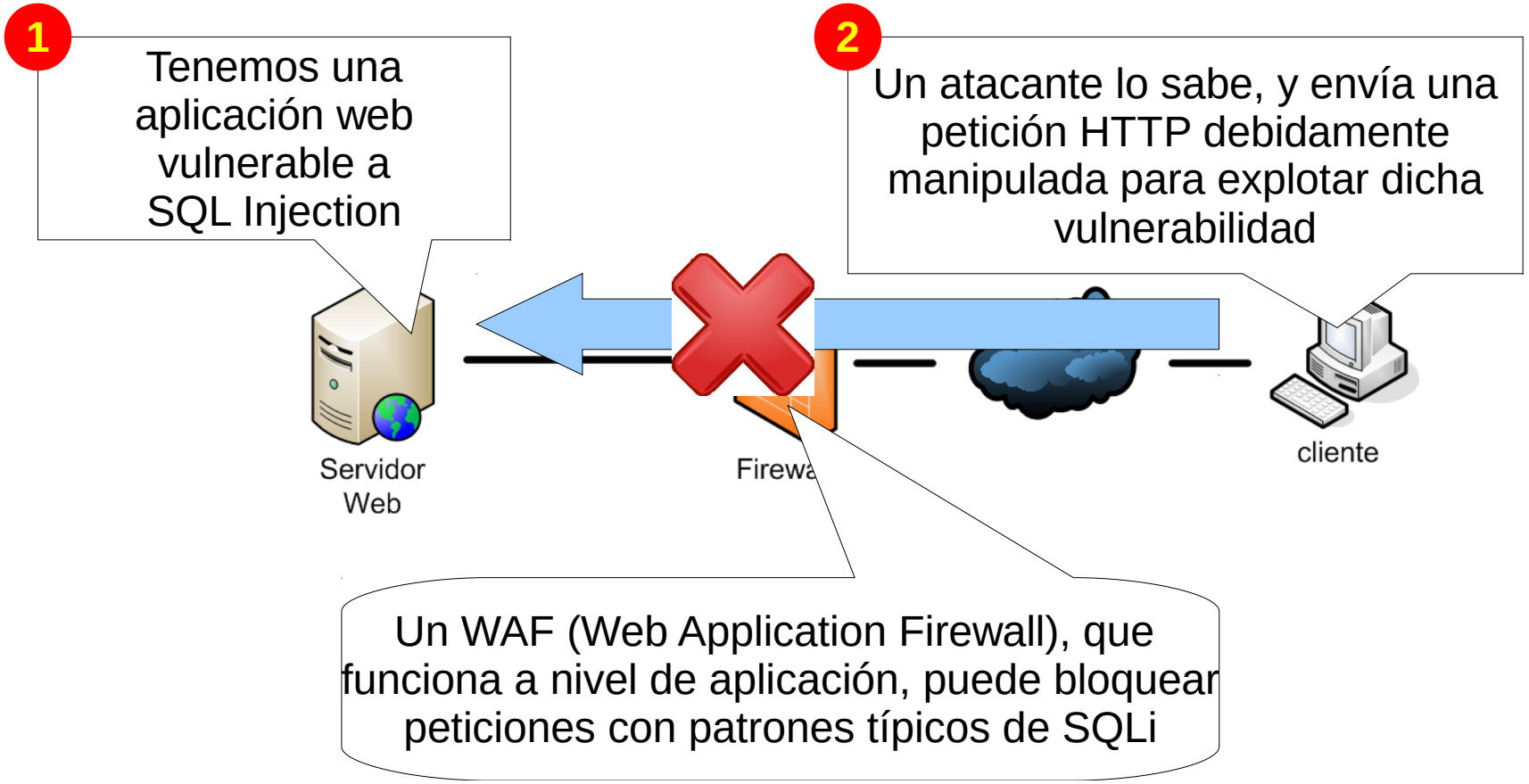
- Filtrado a nivel de aplicación
  - Examinan el contenido del paquete, no sólo las cabeceras IP y de capa 4
  - Mayor capacidad de análisis
    - Mejor gestión de conexiones para ciertos protocolos (ej: FTP)
    - Escaneo de virus y *malware* en general
    - Bloqueo de cierto tipo de ataques a nivel de aplicación (ej: SQLInjection)
    - Bloqueo de ciertos contenidos: *applets* Java, ciertos tipos MIME...
    - Bloqueo de ciertos comandos del protocolo de aplicación (ex: HTTP DELETE)
  - También pueden emplear información de las capas inferiores
    - Ej: origen y destino, estado de conexión...

Es un tipo de firewall más inteligente que no solo mira los datos básicos como la IP o el puerto, sino que abre el paquete y lee lo que hay dentro, como si leyera el mensaje completo.

Filtrado de Capa de Aplicación



Filtrado de Capa de Aplicación



Filtrado de Capa de Aplicación

- Ventajas
  - Mejor control de conexiones para ciertos protocolos
    - Es habitual que firewalls que son fundamentalmente de filtrado de paquetes incorporen conocimiento a nivel de aplicación para ciertos protocolos (ex: FTP)
  - Identificación de ataques a nivel de aplicación
  - Mayor capacidad de *logging* : permite tener un registro con info de ip, puerto y contenido
  - Identificación de protocolos a nivel aplicación
    - Ej: *malware* que utiliza puerto 80 para enviar información a un servidor remoto

Filtrado de Capa de Aplicación

- Limitaciones
  - Soporte limitado de aplicaciones y protocolos -> solo funciona bien para protocolos comunes, no entiende bien de protocolos nuevos
    - El filtrado a nivel de aplicación es complejo
    - Normalmente sólo para ciertos protocolos. Ej: HTTP
    - Problemas con protocolos nuevos o propietarios
  - Menor rendimiento -> más lento, consume muchos recursos al tener que leer todo el contenido
    - Deben analizar el contenido del paquete
    - Problemas con aplicaciones de tiempo real o que precisen un gran ancho de banda
  - Siguen sin resolver el problema de la autenticación a nivel de usuario -

Aunque analiza el contenido sigue sin saber quien es la persona que hay detrás de la conexión.
Solo ve IPs, no nombres de usuario y estas ips puede ser de redes externas, vpns etc

Filtrado de Capa de Aplicación

Tipo firewall	Ventajas	Limitaciones
Filtrado de paquetes (sin estado)	- Eficiencia	- No gestionan conexiones - No evitan ataques a nivel de aplicación - No autentican usuarios
Filtrado de paquetes (con estado)	- Eficiencia - Gestión de conexiones para mayoría de protocolos	- No evita ataques a nivel de aplicación - No autentican usuarios
Filtrado a nivel de aplicación	- Mejor control conexiones - Evitan ciertos ataques a nivel de aplicación	- Menor rendemiento - Soporte de protocolos limitado - No autentican usuarios

Tecnologías Firewall

2.4- Traducción de Direcciones

Network Address Translation

- NAT Estático
- NAT Dinámico
- PAT

Network Address Translation

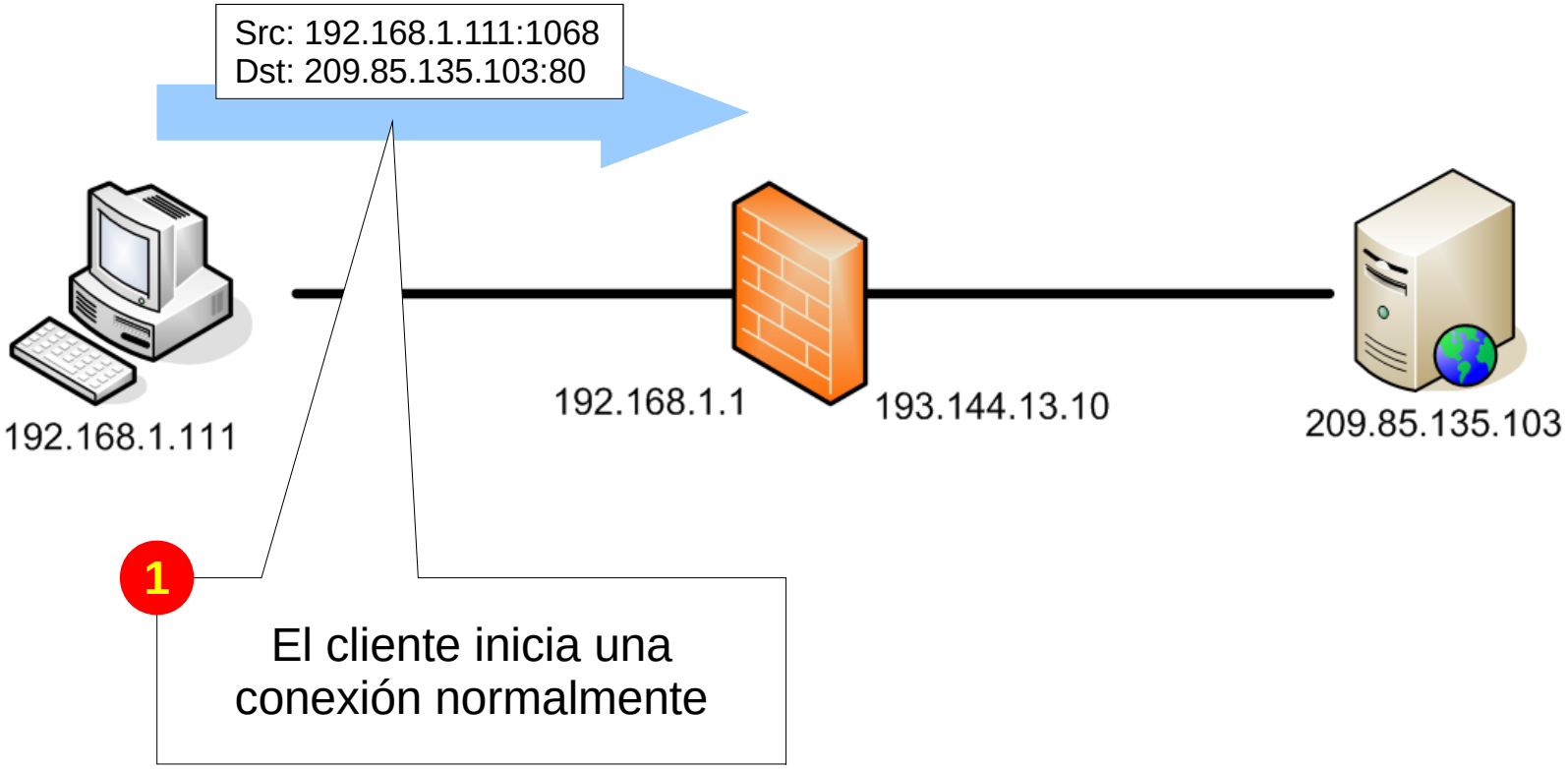
- NAT: Network Address Translation
  - Separa direcciones IP internas de externas

- Ventajas
  - Ahorro de direcciones IP mediante o uso de IPs privadas (RFC 1918)
  - Ayuda a garantizar el control del firewall sobre las conexiones con el exterior
  - Ayuda a ocultar el esquema de direccionamiento interno
  - Ayuda a restringir el tráfico entrante

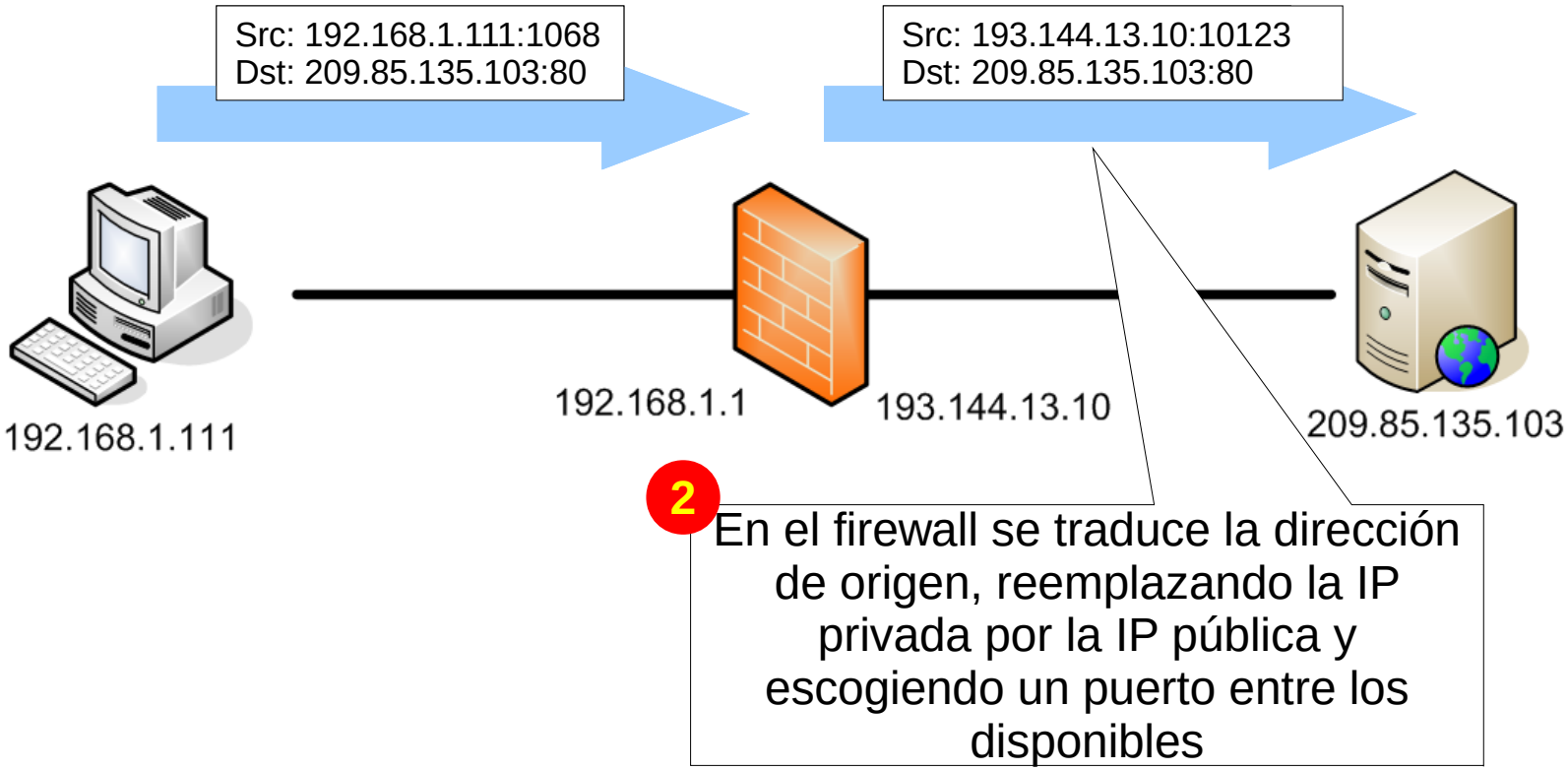
Network Address Translation

- Tipos
  - NAT Estático
    - Cada IP privada tiene asociada una IP pública
  - NAT Dinámico
    - Conexiones basadas en un “pool” de direcciones IP públicas
  - Port Address Translation (PAT)
    - También conocido como NAT con sobrecarga
    - Se utiliza una IP pública única
    - *Mapeo* basado en el número de puerto
    - Muy usado en entornos firewall e incluso en el ámbito doméstico

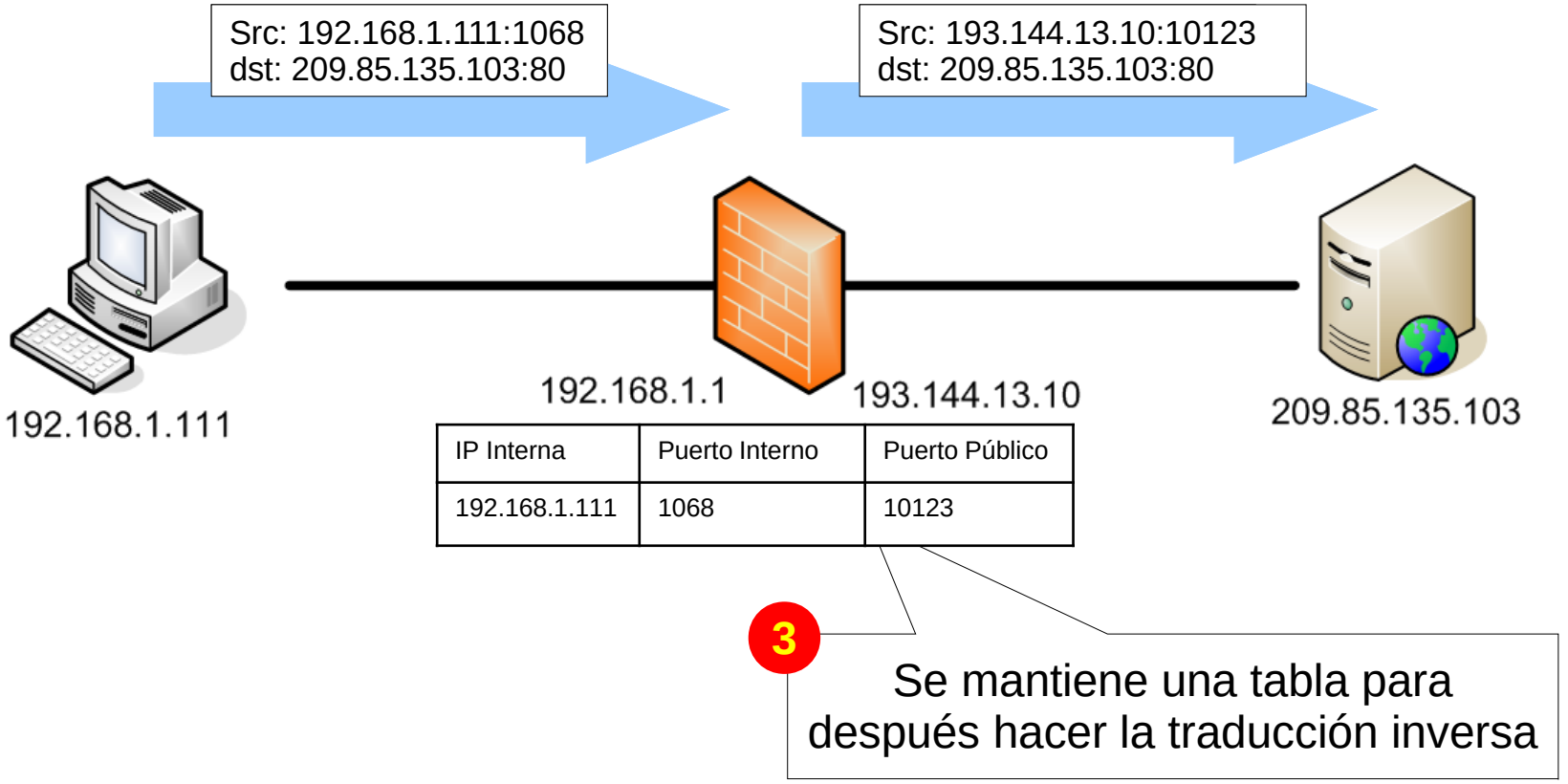
Network Address Translation: PAT (Funcionamiento)



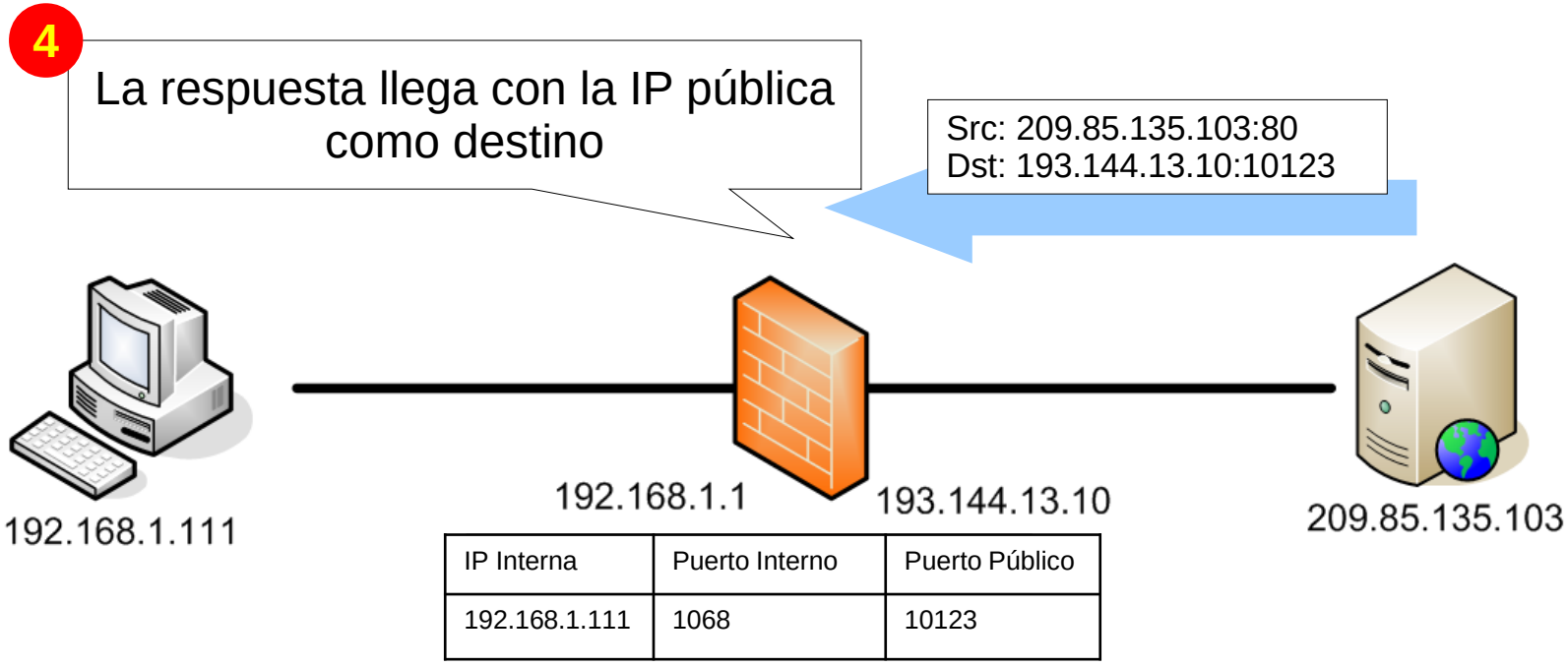
Network Address Translation: PAT (Funcionamiento)



Network Address Translation: PAT (Funcionamiento)

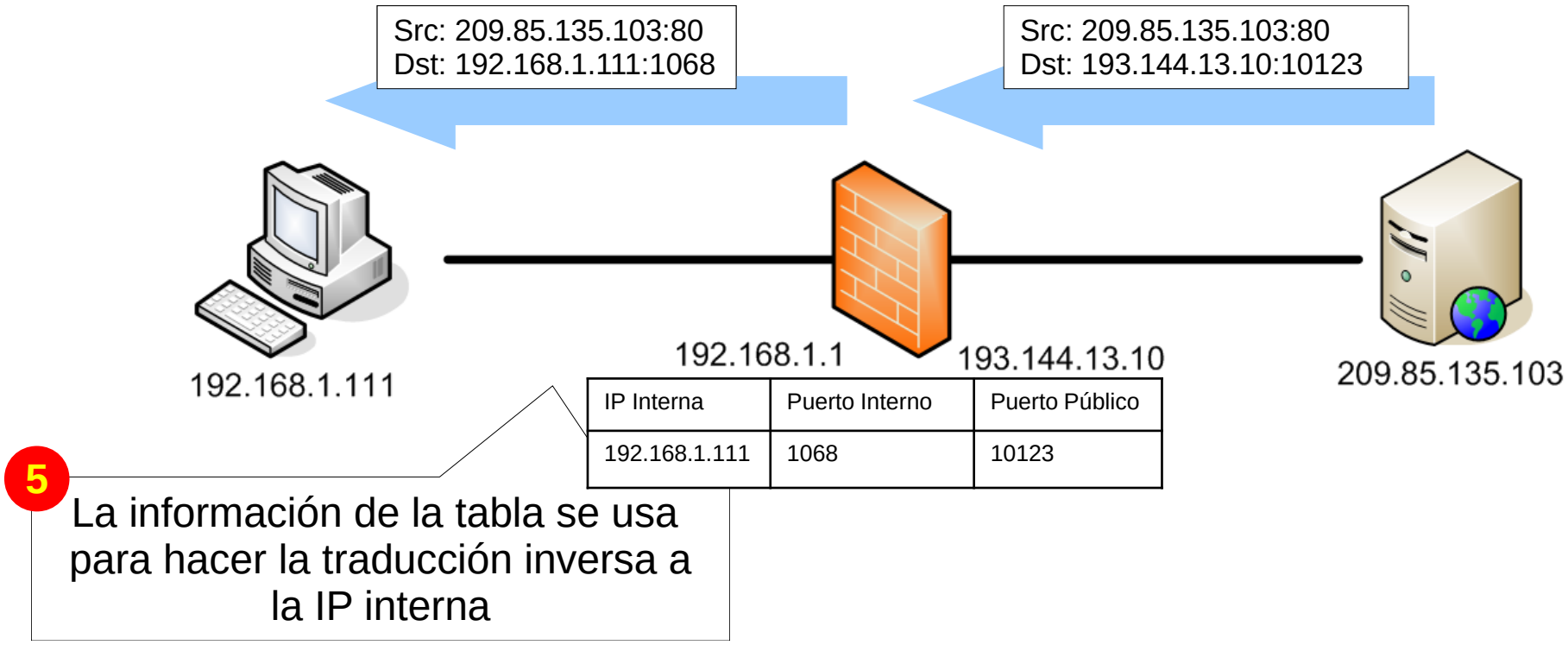


Network Address Translation: PAT (Funcionamiento)



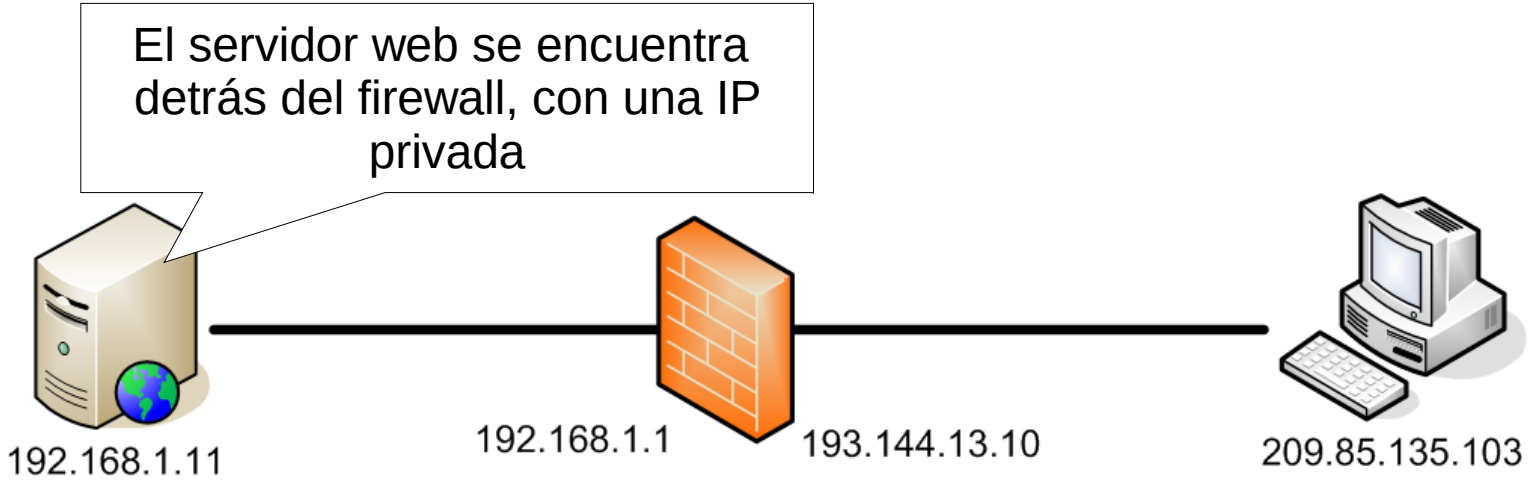
IP Interna	Puerto Interno	Puerto Público
192.168.1.111	1068	10123

Network Address Translation: PAT (Funcionamiento)



Network Address Translation

- También se permite el acceso desde el exterior a máquinas internas

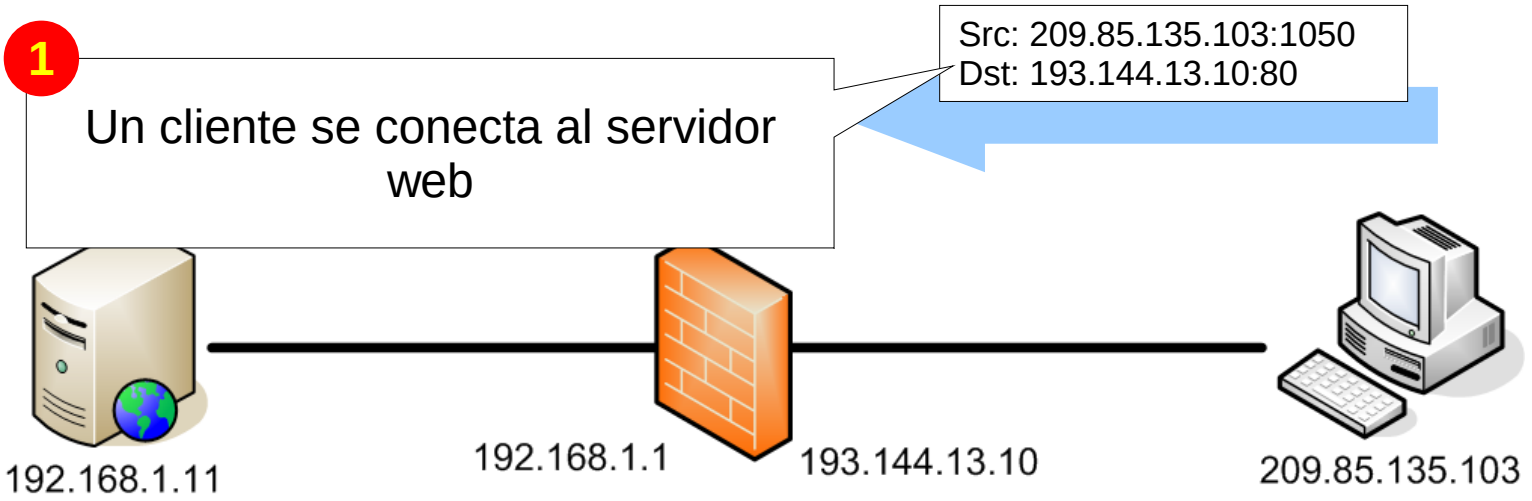


Una entrada en la tabla de traducción del firewall indica que el puerto 80 de la interfaz pública debe mapearse al puerto 80 del servidor web interno

IP Interna	Puerto Interno	Puerto Público
192.168.1.11	80	80

Network Address Translation

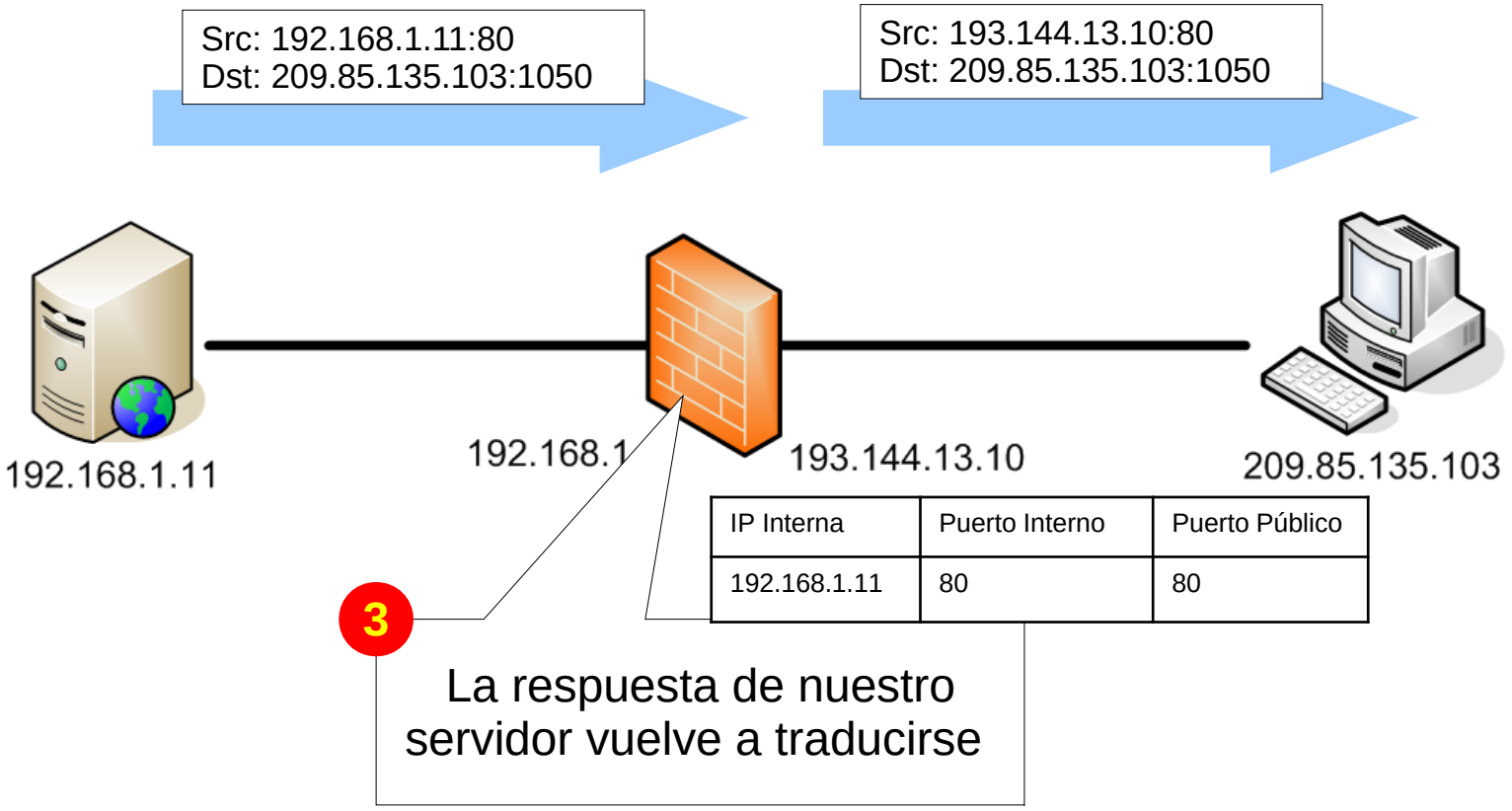
- También se permite el acceso desde el exterior a máquinas internas



IP Interna	Puerto Interno	Puerto Público
192.168.1.11	80	80

Network Address Translation

- También se permite el acceso desde el exterior a máquinas internas



3- Seguridad Perimetral: Política de Seguridad

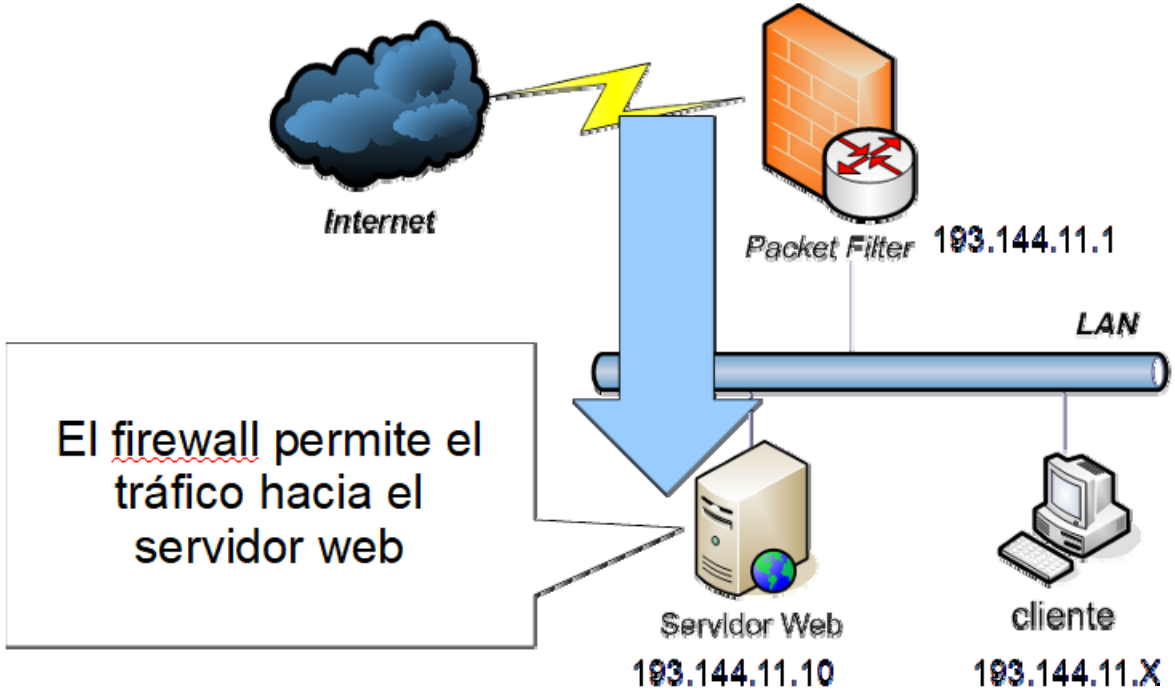
Seguridad Perimetral

- Una arquitectura de red basada en seguridad perimetral establece que los diferentes segmentos de red se organizan en base a zonas de seguridad.
  - En la zona interna (o red interna) se sitúan los usuarios de la propia organización.
  - En la zona externa (o red externa) se encuentran los dispositivos cuya gestión está fuera del alcance de la organización.
  - En la DMZ (*De-Militarized Zone*) se suelen situar a los servidores corporativos que proporcionan servicios a los puestos de la red interna y a clientes situados en la red externa

Seguridad Perimetral

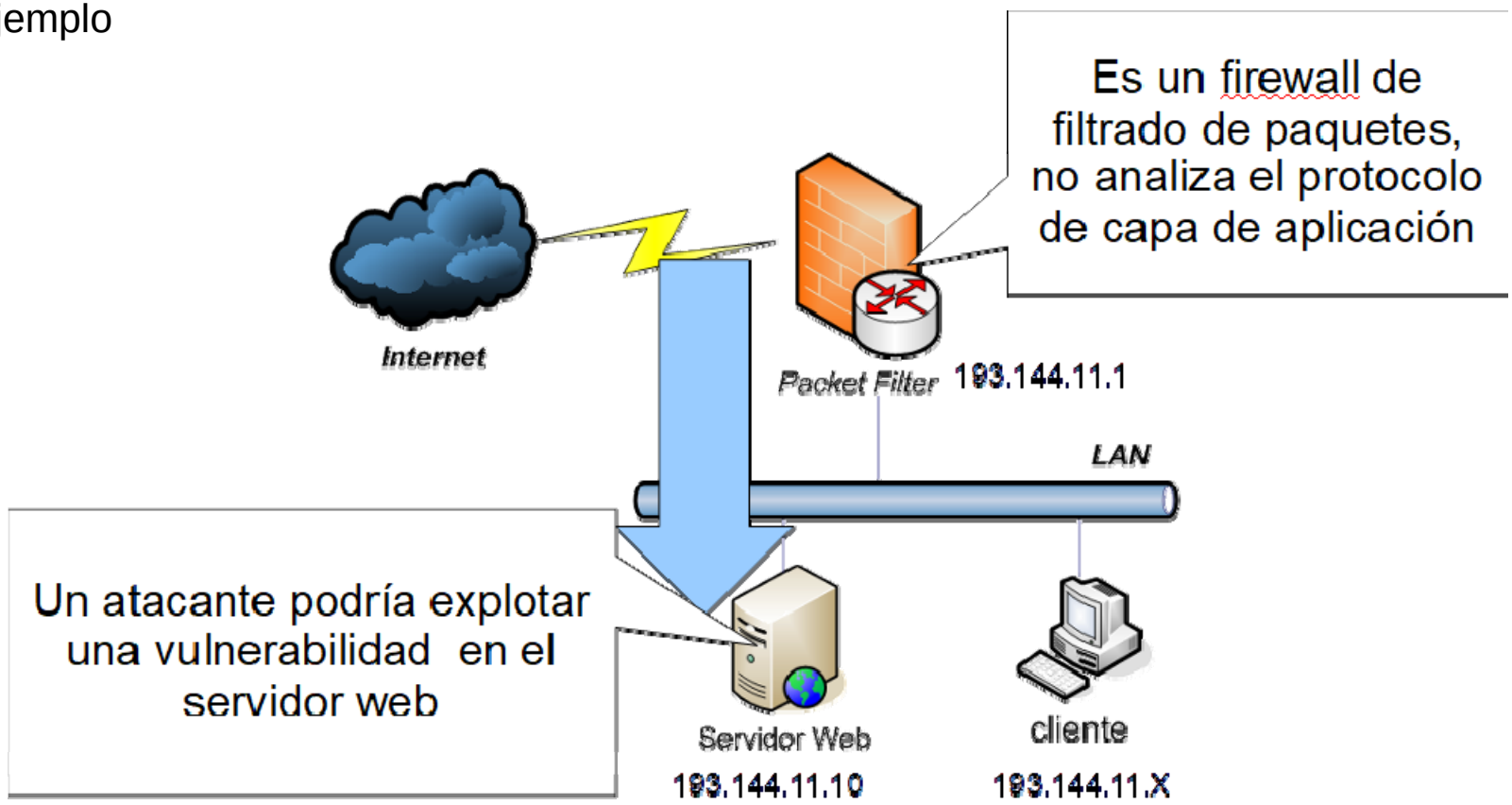
- El intercambio de tráfico entre las diferentes zonas de seguridad están controlados por un *firewall,* en el que se aplican diferentes reglas de seguridad implementadas, por ejemplo, por medio de listas de control de acceso
  - El *firewall* sólo puede bloquear tráfico que pasa por él
  - El diseño de la red tiene una gran importancia en la eficacia de un *firewall*

- Ejemplo:



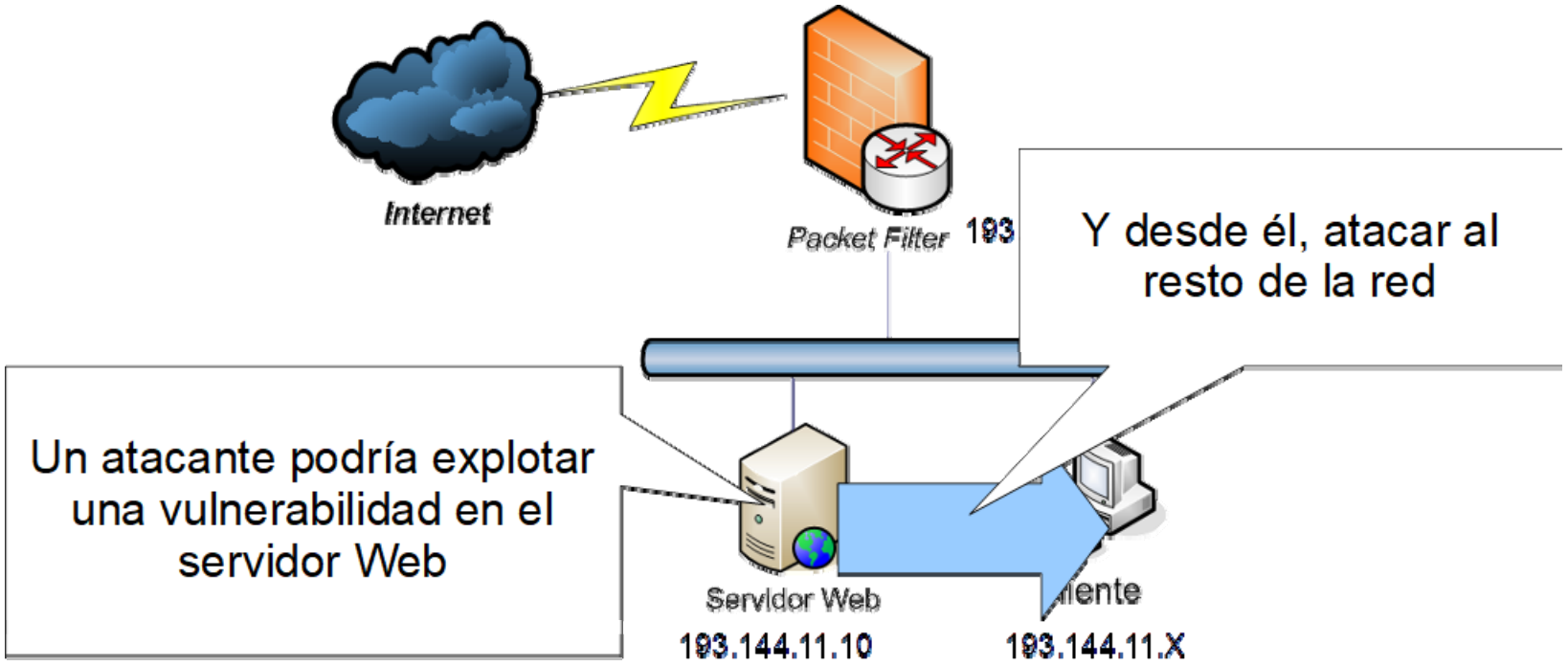
Seguridad Perimetral

- Ejemplo



Seguridad Perimetral

- Ejemplo:

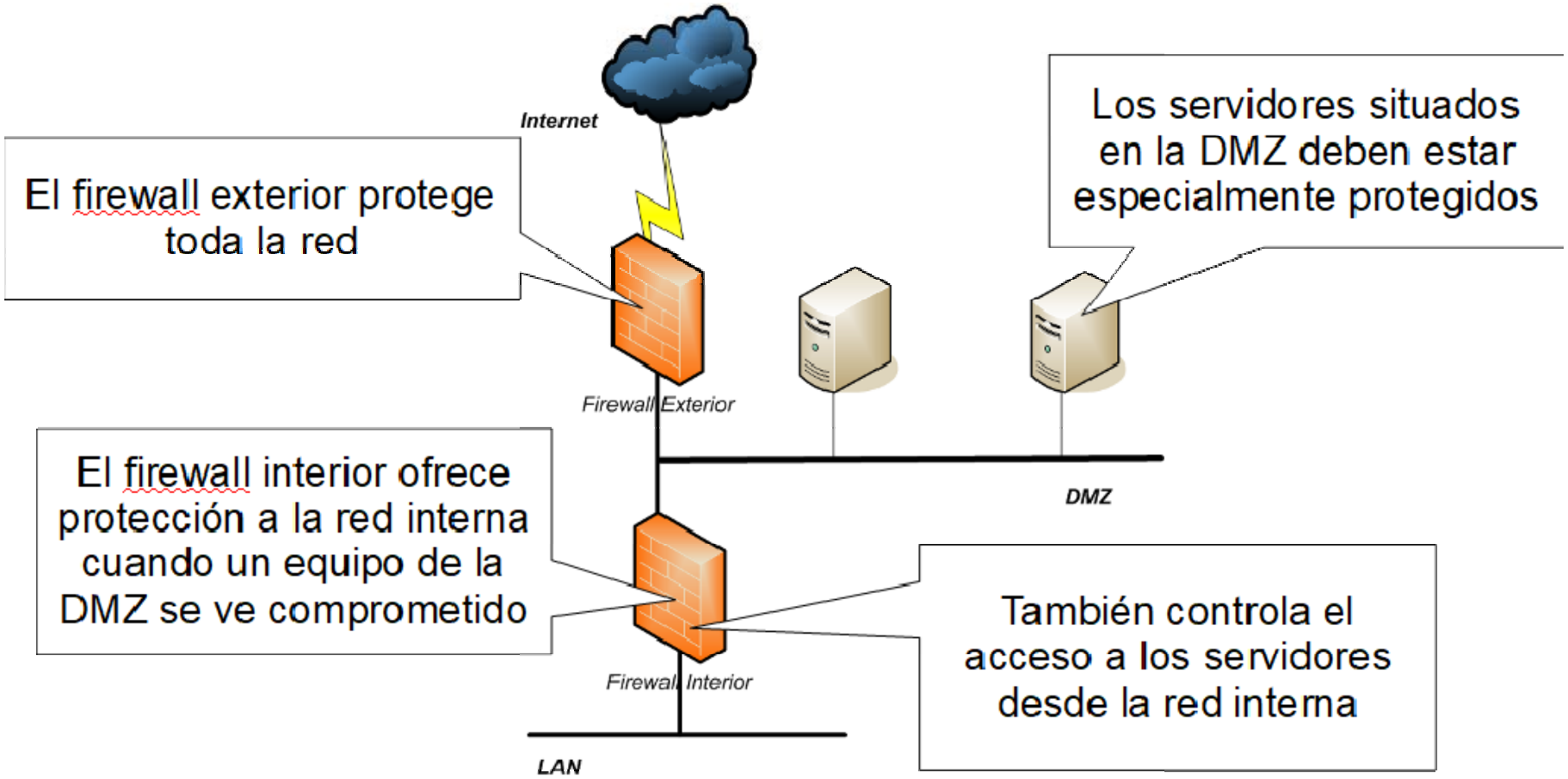


Seguridad Perimetral

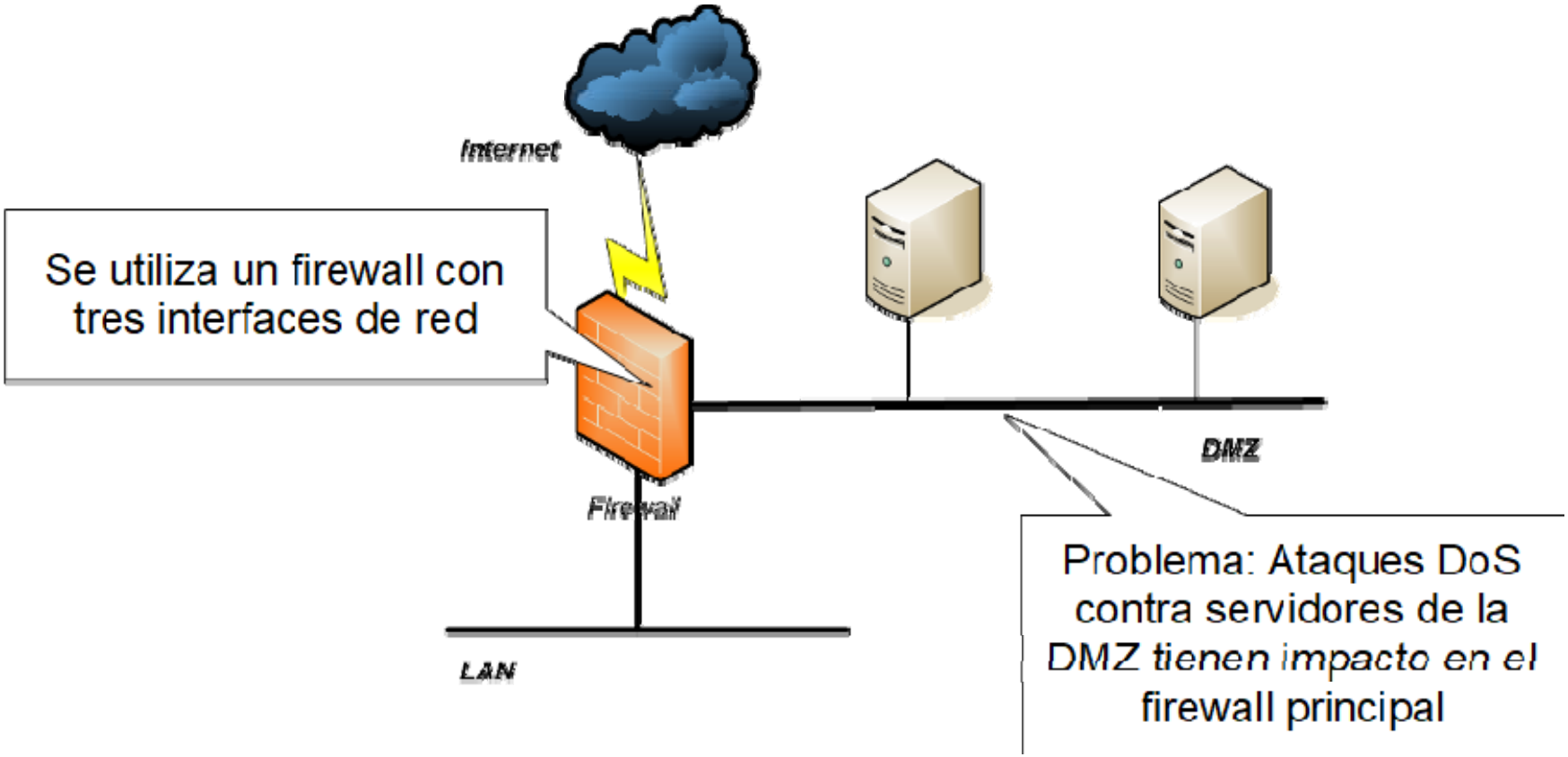
- Entornos *Firewall*
  - Conjunto de sistemas y componentes implicados en ofrecer la funcionalidad completa de *firewall* en una red: filtrado de paquetes, *proxies*, topologías de red específicas...

- Recomendaciones
  - Simplicidad
  - Usar dispositivos para aquello que fueron diseñados
  - Conocer aquello que se está protegiendo
  - Defensa en profundidad
  - Menor privilegio posible
  - Cuidado con las amenazas internas

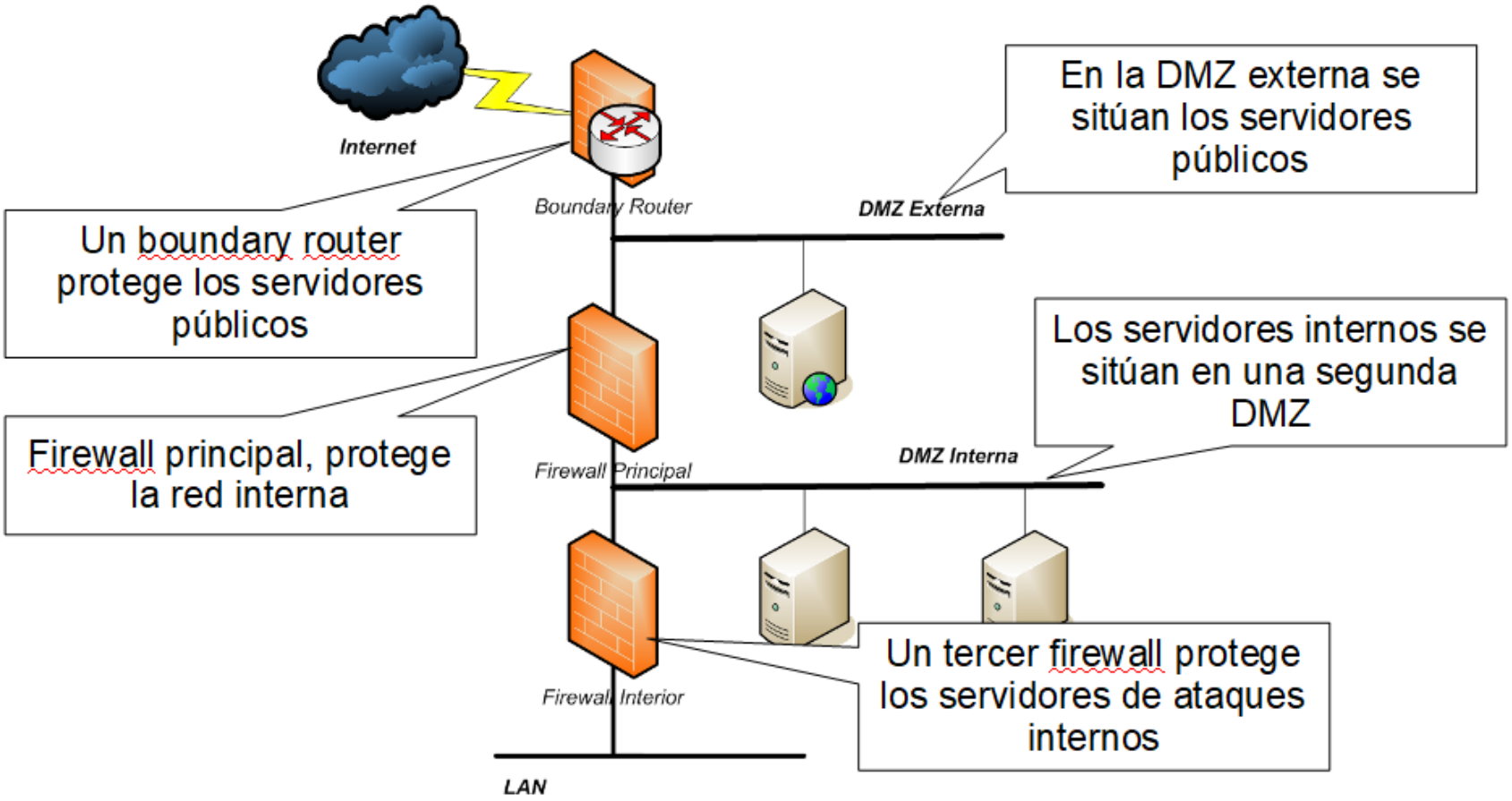
Implementaciones: arquitectura de doble *firewall*



Implementaciones: arquitectura *service leg*



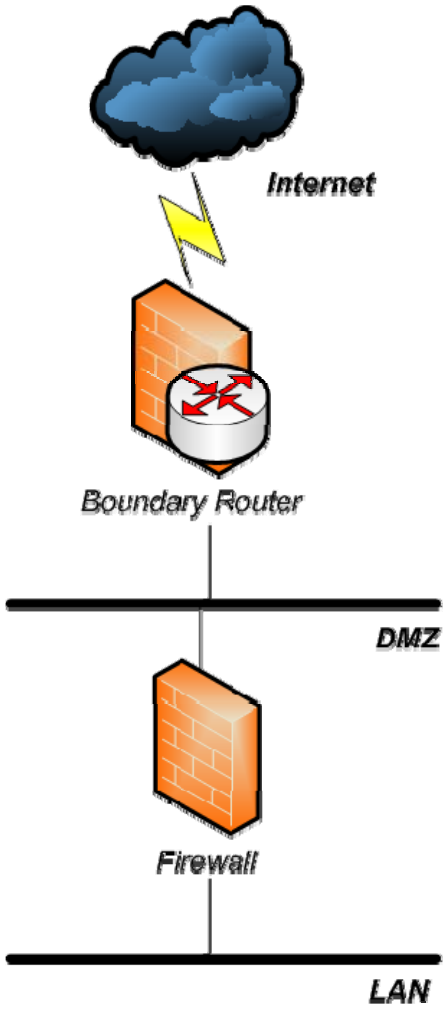
Implementaciones: arquitectura de doble DMZ



Implementaciones: el *router* frontera

- Un *firewall* de filtrado de paquetes es ideal como protección de frontera
  - Control de acceso simple y eficiente
  - Bloquea protocolos no deseados
  - Evita un buen número de ataques sin tener un impacto elevado en el rendimiento de la red

- Uso de un segundo *firewall* con capacidades más avanzadas
  - Examen más exhaustivo del tráfico
  - Diferente política



Otros elementos de seguridad

- *Virtual Private Networks (VPN)*
  - Utilidad en entornos firewall
    - Acceso de usuarios desde fuera de la red protegida
    - Conexión de redes a través de un medio inseguro como Internet
  - El servidor VPN suele situarse en el *firewall*

- *Intrusion Detection Systems (IDS)*
  - Suelen formar parte de entornos *firewall*
  - Situados en las distintas redes protegidas y DMZs, determinados hosts...
  - Algunos incluso interactúan con estos, por ejemplo para bloquear intentos de intrusión (IDS reactivos / IPS)

Ubicación de los servidores

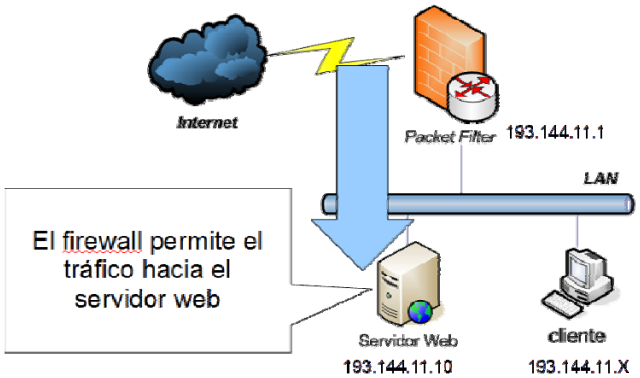
- ¿Dónde situar los servidores? Factores a tener en cuenta:
  - Número de DMZs
  - Acceso necesario (interno/externo)
  - Volumen de tráfico
  - Importancia del servidor y de los datos que contiene

- Consideraciones generales
  - Proteger servidores externos mediante *packet filters*
  - No situar servidores externos en la red interna
  - Proteger los servidores internos con un *firewall* interno según la importancia de estos o las restricciones de acceso
  - Aislar servidores para evitar poner en peligro al resto de la red ante ataques sobre ellos

Política de Seguridad

- Determina cómo se maneja en el firewall el tráfico de red, así como los procedimientos de administración y actualización del mismo
- Paso previo a establecer una política de seguridad: análisis de riesgos en las aplicaciones necesarias en la organización
- Pasos necesarios para la creación de una política de seguridad en el *firewall*:
  - Identificación de las aplicaciones de red necesarias
  - Identificación de las vulnerabilidades relacionadas con cada aplicación
  - Análisis de coste-beneficio de los métodos empleados para asegurar cada aplicación
  - Análisis de tráfico entre aplicaciones (matriz de tráfico)
  - Definición de las reglas del *firewall* a partir de la matriz de tráfico entre aplicaciones

Política de Seguridad: matriz de tráfico



Origen / Destino	193.144.11.0/24	0.0.0.0/0
193.144.11.0/24	-	Permitir la navegación Web
0.0.0.0/0	Permitir el acceso al servidor Web que está en 193.144.11.10 Permitir el tráfico de retorno de las peticiones Web realizadas desde el interior	-

Política de Seguridad

- Otras consideraciones:
  - Verificación de la política de seguridad
  - Administración y mantenimiento del *firewall*
  - Revisión periódica de las políticas de seguridad
    - Auditorías de seguridad
    - Actualización de políticas de seguridad
  - Acceso al *firewall*
  - Política de *logging*